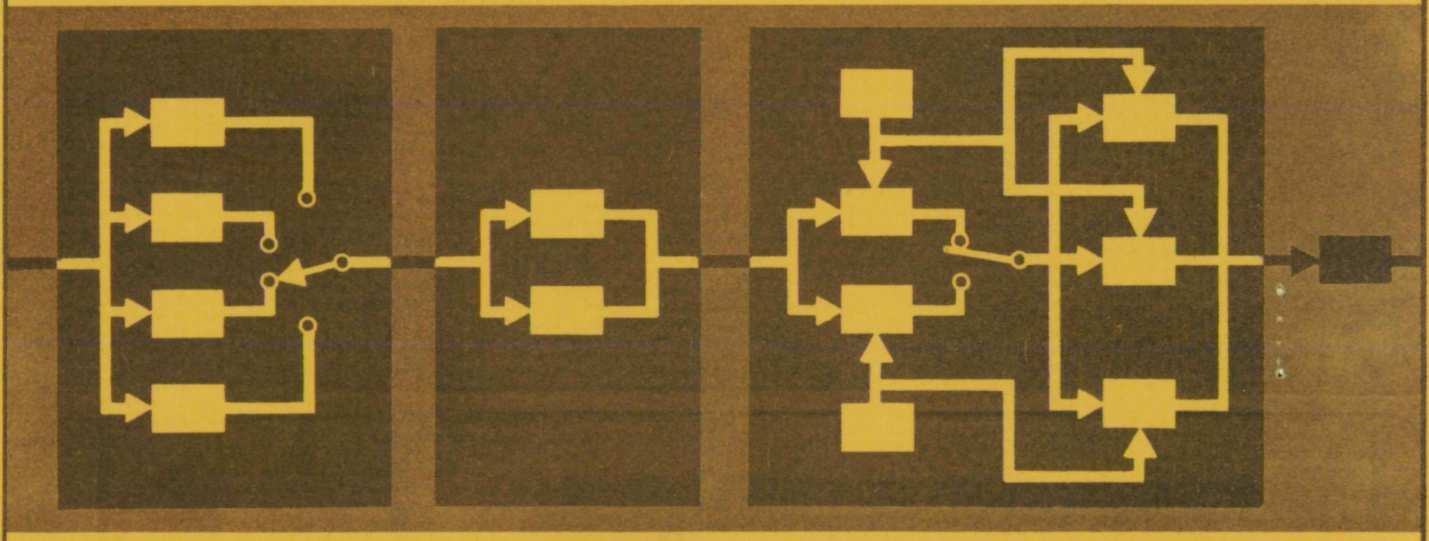
7620-71-001
July 1971

HANDBOOK

DESIGN OF AUTOMATED REDUNDANCY VERIFICATION



Prepared for
JOHN F. KENNEDY SPACE CENTER, NASA
Kennedy Space Center, Florida 32899

Prepared by
Radiation Systems Division
Surface Systems Operations
Systems Engineering Department

RADIATION INCORPORATED Systems Division MELBOURNE, FLORIDA 32901

SUBSIDIARY OF HARRIS-INTERTYPE CORPORATION

7620-71-001

HANDBOOK

DESIGN OF AUTOMATED REDUNDANCY VERIFICATION

JULY 1971

Prepared by

SYSTEMS ENGINEERING DEPARTMENT
SURFACE SYSTEMS OPERATIONS
RADIATION SYSTEMS DIVISION

Principal Contributors

F. A. FORD
T. W. HASSLINGER
F. J. MORENO

Prepared for

JOHN F. KENNEDY SPACE CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Under Contract Number NAS10-7420

FOREWORD

There is an ever increasing need for the capability of assessing the operational integrity of redundant items in a system on an automated and noninterfering basis. This need saw its genesis in the past decade of space exploration when the use of redundancy burgeoned as a design technique. However, not until the prospects of reusable space vehicles such as Space Shuttle became a contractual reality did the need emerge as a prominent problem.

Attendant to recognition of the problem is the necessity of identifying solution methods and means of assessing the impact of this added capability on otherwise acceptable designs. It is to this end that this handbook is directed.

Content of the handbook has been determined to a great extent by the availability of other related references suitable for direct application. Such references are quite limited. At first glance, the problem may appear to be a simple extrapolation of existing Automatic Checkout Equipment technology. However, the requirement of noninterference severely limits these benefits and, consequently, much of the precedence on which to build. To be an effective reference, then, the handbook must be more than a digest of related principles. Emphasis has been placed on solution techniques and procedures using step-by-step development for the application of the methods discussed. To assist the designer in coarse design planning, a section of guidelines, pitfalls, do's and don'ts has also been included to cover typical cases.

Finally, the reader desiring a familiarization with the subject should find it necessary to read only the Introduction (Section 1.0) and the Overall Redundancy/ Redundancy Verification Design Guidelines (Section 8.0).

T.W.H.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

1.0 INTRODUCTION

This handbook provides a unified approach to the design of automated redundancy verification. These introductory remarks are intended to place the approach in context, describe the use of the handbook and provide the design process overview. The discussion begins with a description of the redundancy verification problem (Section 1.1). To facilitate rapid identification, the problem has been illustrated with examples portraying why automated redundancy verification is necessary and the types of situations to which it typically applies. In short, these examples are intended to identify the types of problems to which this handbook is to be applied.

Section 1.2 provides a more definitive description of the handbook purpose and delimits the scope. Section 1.3 describes how to make efficient use of the handbook with explanations of its features and organization. Finally, Section 1.4 provides an overview of the process used to develop an automated redundancy verification design.

1.1 THE REDUNDANCY VERIFICATION PROBLEM

It is appropriate to ask why one should be concerned about redundancy verification. Why should there be concern over how much redundancy is present so long as the system is performing its function? Consider a system which is intended to perform a mission of a predetermined length. The system will have an inherent probability of completing this mission without loss of predefined functions. Should the system contain redundancy, this probability will undoubtedly be based on the assumption that the redundancy was present at the start of the mission. Is this a valid assumption when the system has had time logged prior to the mission? The answer obviously depends on the amount of time logged prior to the mission, the length of the mission, the reliability of the equipment in question and the criticality of the mission. In any event it seems prudent to verify the existence of redundancy. It is important to note that this same situation exists for a newly purchased one-shot system such as a Saturn V or a system which will perform many missions over its lifetime such as a fighter plane.

Next consider a system such as a spacecraft which has several distinct phases in its mission, each representing successively greater commitments in terms of risk and fuel consumption. Before commitment is made to the next phase, an appraisal is made of the likelihood of completing that phase in terms of the present condition of the system (and its crew). If the spacecraft contains redundancy (and the chances are slim that it does not), the condition or *status* of this redundancy will likely be determined before further commitment is attempted. Note that, in contrast to the first situation which addressed redundancy verification at the start of the mission, this situation is concerned with verification during the mission. The two are fundamentally different in terms of an approach to verification. In the former instance, most of the verification equipment could be designed as extrasystem check hardware. In the latter situation much of the verification equipment must fly with the spacecraft (or be allocated telemetry channels for remote analysis and verification). It is additionally worth noting that this situation might also apply to a system that is capable of aborting its mission at any time (as contrasted to the discrete commitment levels described). If this decision were based on the amount of redundancy present to allow a "safety factor," the verification would likely be on a continuous or real time basis.

A third prominent situation illustrating the use of redundancy verification is related to the maintenance policy of a system. Consider a system which has, for all practical purposes, an indefinite mission such as a communications relay link. Assume that a maintenance crew is on duty at all times and further recognize that the system contains switched redundancy to enable maintenance to be performed on off-line items without disrupting system performance. If redundancy verification is employed such that a *real time* indication of the status of each redundant item is available, maintenance can be effected immediately following a failure. While maintenance is being performed on one item, the redundant item is continuing to perform the desired function, thus allowing the system to continue with the mission. Such maintenance policies are frequently used on large ground systems and always require some form of automated redundancy verification. The importance of the approach can best be appreciated from a reliability standpoint. Reference [1] shows that this form of redundancy, i.e., maintained redundancy, realizes the most significant gain in reliability.

In the above situation, it may be desirable to exploit the results of verification even further. Why not use the status indications to drive switches and thus provide automatic reconfiguration of the system in the event of a failure? This would provide rapid recovery from a failed condition and is often utilized for systems operating unattended. Under these circumstances, it is not sufficient to simply know that a failure has occurred somewhere in the system. The specific item which failed must be identified so that it may be switched out and the redundant item switched in to take its place. At this point, such a statement may seem to be an obvious conclusion. It will be shown in Section 3.0, however, that this deceptively simple statement penetrates the core of verification design.

From the illustrations above, it is possible to form a general statement of the verification problem. Examining the differences first, it should be obvious that the final solution is quite dependent on the mission or operations considerations. These determine: a) if real time or deferred time verification is required, b) whether a specific piece of equipment must be identified as having failed or simply the indication given that a failure has occurred *somewhere* and c) while not obvious from the illustrations, the degree of confidence in the verification results.

Looking at the points in common among the illustrations, each instance of verification involves an automated means of indicating the operational integrity of the *Item Being Verified* (IBV). Furthermore, the indication of operational integrity in each illustration was discrete in nature, i.e., redundancy was present or not, the item was working or not, the redundancy complement has degraded from three available to two available. In general, the indication would take the form of statements such as go, no-go or green, amber, red. This discrete indication of operational integrity is called status. *The objective of redundancy verification is then to indicate status of IBV's.* In the case where it is desired to indicate whether redundancy is present or not, the IBV would be the entire redundant network and status indications could be simply "present" or "not present."

To this point, the illustrations have served to exemplify the possible diversity in requirements and objectives of redundancy verification. Nothing has been said about how the indication of operational integrity (in terms of status) is achieved. If status is to be determined on an automated basis, clearly, some *performance property* or properties of the IBV must be identified and measured. Identifying and extracting these measures constitutes the largest single problem in redundancy verification. The measures will typically be obtained from physical properties of input and output signals and/or effects (voltage,

pressure, angular travel, light intensity, etc.). *In general terms, the redundancy verification problem is then one of determining IBV status from physical properties of input and output signals/effects.*

1.2 PURPOSE AND SCOPE

This handbook has been prepared with the principal aim of aiding the designer in planning for, developing, assessing and implementing automated redundancy verification. To this end, the handbook provides:

- descriptions of the design process,
- step-by-step design considerations with appropriate analysis techniques,
- ample tutorial material on the implementation of the methodology,
- design aids in the form of graphs and charts for ready reference,
- an example illustrating the use of each design aid in addition to application examples of the overall process, and
- overall design guidelines and identification of general practices to be adhered to and to be avoided.

The secondary aim of the handbook is to indicate the impact of redundancy verification on the overall design approach to a system. This will enable the systems designer to forecast and plan for the needs of redundancy verification during the early stages of system development.

The handbook is not and cannot be a complete guide to redundancy verification. To approach this ambitious endeavor would require several volumes. The situations which should be most frequently encountered have been amply covered. Special applications and design formulations must be treated individually. References have been cited to aid the designer in this regard as the handbook is not meant to replace imagination but to foster it.

The problems of verifying operation of redundancy verification equipment or verifying redundancy within redundancy verification equipment have *not* been addressed. This is not meant to imply that these problems are insignificant, only that space does not permit their treatment. Both problems fall into that time honored avalanche of "checking the checker. . . ."

As indicated in Section 1.1, one motivation for using redundancy verification is to drive automatic reconfiguration circuitry. The implications of this application are identified but the design of automatic reconfiguration is *not* discussed.

1.3 USE AND ORGANIZATION OF THE HANDBOOK

The handbook can be thought of as being divided into four major subject areas as illustrated in Figure 1.3–1. Key topics are also shown in the figure. When used with the page numbers at the right, the key topics in Figure 1.3–1 serve as a guide to locating subject areas.

Subject Area	Key Topics	Section	Page Numbers
I. Design Process	• Steps in the development of a redundancy verification design • Description of each step • Analysis techniques • Design aids and examples • Design interrelationships	1.4, 2.0 – 5.0	7 – 116
II. Additional Considerations	• Cost • IBV isolation devices (switches, relays and diodes)	6.0	117 – 120
III. Examples	• Illustrate overall design process • Design critiques for redundancy and redundancy verification	7.0	121 – 140
IV. Design Guidelines	• Do's and Don'ts • Practices to be adhered to and avoided • Rules of thumb for planning	8.0	141 – 152

Figure 1.3–1. Handbook Subject Reference

The area covering the design process constitutes the bulk of the handbook. In effect, this group of sections describes how a design is realized and at the same time develops some of the special techniques, or special applications of familiar design principles, necessary for redundancy verification. The entire process is linked pictorially by a network of flow diagrams which trace the progress of the design. An indenturing or subordination technique is used to enhance the organization and presentation. This is depicted in Figure 1.3–2. A top level diagram is used to portray the design overview. Blocks in this diagram which require more detail are expanded into second level diagrams. Each block is assigned an unique alphanumeric identification using the same basic scheme for section numbering in technical reports. To reduce confusion with figure and table references, top level blocks are assigned letter identifiers. That is, blocks in the overview diagram will have letters ending in decimal-zero, e.g., E.0, G.0. Subtier blocks will bear the letter of the overview block but not end in zero, e.g., E.1, E.7 are subordinate blocks to overview block E.0. Each block also identifies the section number(s) in which it is discussed. To complete the cross referencing, relevant block identifications are indicated at section headings. It is impossible to be consistent in the block numbering scheme and at the same time have block numbers agree with section numbers in the text. Therefore, no attempt is made to correlate block numbers with section numbers. Also, block numbers do not necessarily represent order in the design process. As an additional orientation aid, caption sheets at each major section contain the segment of the overview diagram discussed in that section.

The sections covering additional considerations and examples are felt to be self-explanatory. The section relating design guidelines is intended to be streamlined for quick reference. The section makes use of stylized information displays, using portions of the material from the design process in summary form with notes on assumptions, restrictions and applicability.

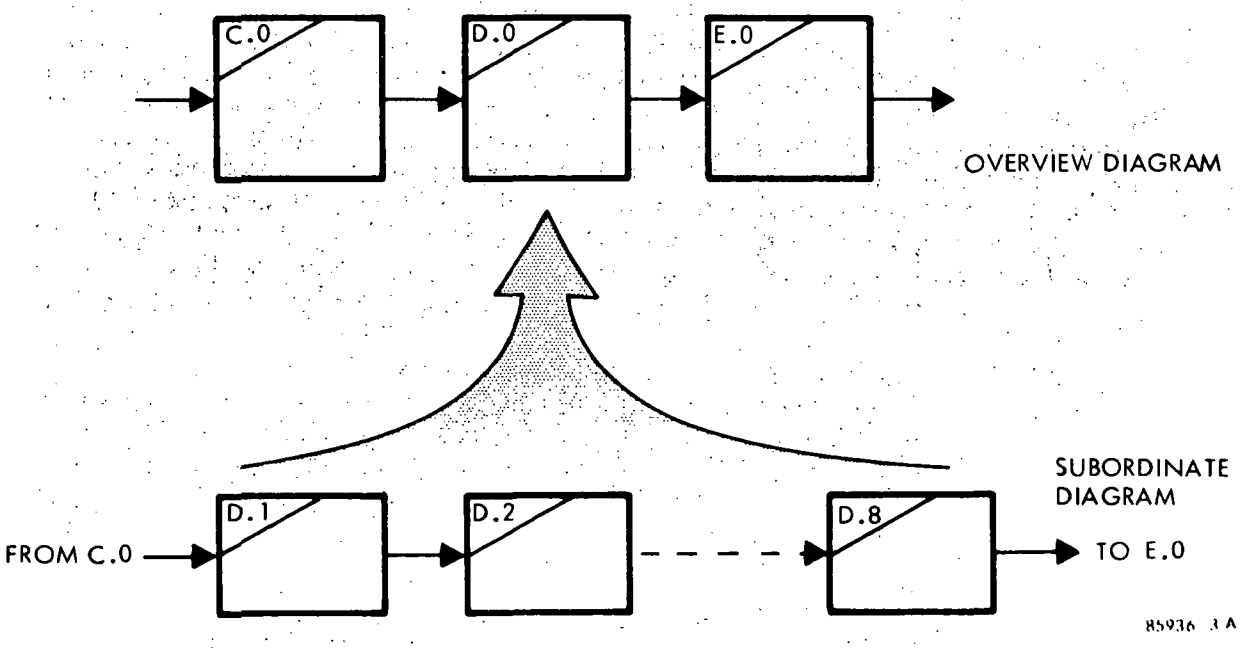


Figure 1.3–2. Relationship of Block Flow Diagrams

To facilitate rapid location of material by section number, the top, outside corner of each page contains the number of the last section discussed on that page. The reader may also find these numbers a convenient reminder of subject organization.

1.4 DESIGN PROCESS OVERVIEW

This section introduces the design process for redundancy verification. The reader not acquainted with the following terms may find it helpful to refer to the Glossary.

Status Development	Status Resolution
Principal System	Status Reporting
Redundancy	Conditional Status
Tenant Signal	Injected Signal

The process is depicted in Figure 1.4–1* with Figure 1.4–2 illustrating relative level of detail addressed by the design as it progresses. So that continuity can be provided between redundancy design and verification design, a brief description of the former is provided. The system which exists at completion of redundancy design, before any verification is added, is called the *principal system* and marks the baseline to which verification is designed. As indicated in Figure 1.4–2, the level of design attention alternates between segments which require overall system consideration and segments which required individual attention to numerous but smaller IBV considerations. Major headings in the sections that follow adhere to the identified areas of design attention.

As indicated in Figure 1.4–1, the design process can be entered from two points. One point corresponds to adding automated redundancy verification to a system already in existence. That is, the principal system is defined and firm. The other point corresponds to the inclusion of redundancy verification into the overall design of a system under development. The major points to be considered when adding verification to an existing system are itemized below.

- The redundancy design cannot be influenced by verification. Some existing redundancy designs may require modification and added verification techniques may be more complex than necessary. It may not even be feasible to add verification.

- The packaging design cannot be influenced by verification. Most applications of verification will require equipment to be colocated with principal system equipment. Finding suitable locations to add hardware, with its associated wiring and grounding distributions, may be quite difficult and expensive to implement.

---

*For the convenience of the reader, this figure is included on a foldout at the end of this section. So that the figure may be referenced simultaneously with other flow diagrams appearing later in the handbook, the figure is duplicated on the back of the foldout.

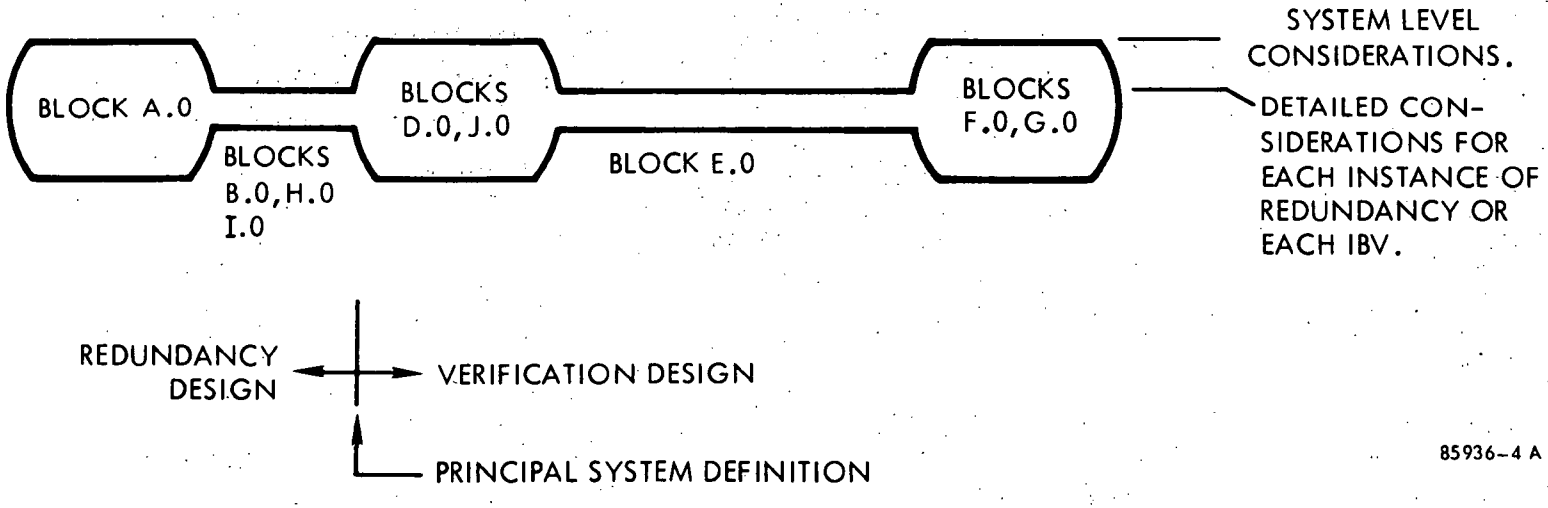


Figure 1.4–2. Progressive Design Detail

- The signal character and content cannot be influenced by verification. Implementation of verification is quite sensitive to the character of existing signals and the approach to injecting known signals for reference. To the extent that information carried by existing signals (called *tenant signals*) is not degraded, it is often expedient to superimpose *injected signals* on tenant signals or to otherwise alter the character of tenant signals to improve verification ability. This may not be feasible if no latitude is available in areas such as IBV bandwidth, signal-to-noise ratio, coding, etc.

- Equipment utilization cannot be influenced by verification. To be cost effective, it is often desirable to share some principal system equipment such as selected secondary power sources, data processors, computers and the like. If the verification design cannot influence compatibility in these areas, an add-on design may have to duplicate the functions.

Having indicated the contrast in the two starting points, attention can now be turned to the individual blocks in Figure 1.4-1. The blocks will be discussed in order of occurrence in the design and not sequentially by block number.

Identification of Design Guidelines and Requirements BLOCK K.0

This block serves to point out factors which do not appear in the immediate development of design flow but which are significant in controlling the end result. As indicated in Figure 1.4–1, these factors pervade every facet of the design and serve as the initial forcing function. Operations requirements and specifications constitute initial problem definition as described by the system funding organization. Design policies and philosophies are established by the system developer during system design and enumerate gross design guidelines, strategies and approaches.

Identify Areas to be Redundant BLOCK A.0

This block is a necessary step in any redundancy design. The handbook will not, however, address the details of its implementation since the selection is very dependent on the individual circumstances at hand. Reference [1], in addition to numerous reliability engineering texts, describes methods for assessing approaches in design trade-offs.

Select Redundancy Design BLOCK B.0

Once areas of redundancy are identified, it will then be necessary to select a design. An abbreviated procedure for achieving redundancy designs is discussed in Section 2.1. Details for implementing the procedure are contained in Reference [1]. One purpose for including this block is to allow the verification designer to become familiar with redundancy design problems and the types of redundancy with which he will be dealing. More important, however, is the necessity of making the redundancy designer aware of additional constraints which must be satisfied if an effective verification design is to be realized. This is the subject of the following two blocks.

Verification Factors Influencing Redundancy BLOCKS H.0 and I.0

These blocks are identified in Figure 1.4–1 as feedback to redundancy design and have been included to indicate factors which a redundancy designer should consider when

automated verification is to be used. There are redundancy designs which cannot be verified and designers should obviously avoid these if at all possible. Probably the most significant factor which the redundancy designer should be aware of is the general tendency of increasing verification error with increasing numbers of IBV's. Increasing the number of IBV's for a given system is the same as dividing the system into smaller pieces. Oftentimes, the redundancy designer is motivated to define smaller pieces as redundant networks since theory indicates that this results in a higher reliability. However, in doing so he will typically increase the probability of error in the verification equipment and defeat the intended purpose.

A second consideration is that of effective decrease in observable system reliability due to the addition of verification. Verification increases one's knowledge of equipment status and the price to be paid is an attendant increase in the number of IBV failure indications due to erroneous indications from verification equipment.

Partition System BLOCK C.0

The function to be performed in this block is straightforward. The task consists of dividing the system functional block diagram into the identifiable items to be verified. This action will also indicate the relationships among the various IBV's, indicate the signal flow and define fixed points for output and possibly input measurements.

Characterize Verification Problem BLOCK D.0

Once the IBV's are identified on a functional basis, it will then be necessary to determine the characteristics of each and to establish design approaches. This includes establishing confidence levels, any signal injection required, whether real or deferred time verification is to be used and detailed status definitions.

Identify Status Development BLOCK E.0

Using the approaches identified in Block D.0, it is now necessary to develop status indications based on IBV performance properties derived from signals and/or effects present at the IBV. The design which takes place in this block holds the key to a successful verification. The tasks consist of identifying properties indicative of operation, extracting and processing voltage analogs of these properties, identifying ranges of the analogs which constitute successful operation and developing a status indication based on these ranges.

Status Development Factors Influencing Characterization BLOCK J.0

This block is indicated as a feedback block in Figure 1.4–1. The purpose of the block is to identify status development factors which may alter previous decisions made during characterization. This alteration is not unusual and design decisions sensitive to subsequent status development findings should be kept in mind by the designer during characterization. The most significant factor is that of confidence as it relates to real time or deferred time verification. It is entirely possible that, after completing a status development design, one may discover the probability of verification error (probability of indicating erroneous status) is too great using real time verification. This obviously results in a low verification confidence. Often, a method of increasing this confidence is to revert to high confidence, deferred time verification. This change will impact the approach identified during design

characterization. Unfortunately, the impact is not always restricted to the IBV in question but may proliferate to others which are functionally related. Quite possibly, the lesson to be learned is to be prepared with a backup approach.

It is important to note that, using some form of deferred time verification, one can always verify redundancy given the fact that it was possible to verify (although at a low confidence) in real time. A designer is motivated to avoid high confidence, deferred time verification, however, since it typically requires an interruption of the principal system mission.

Design Status Resolution BLOCK F.0

Once indications of status have been achieved for each IBV, there arises a new problem associated with functional interrelationships within a system. IBV's will be communicating with other IBV's. Should one fail, it is quite likely that a failure indication will ripple down the chain. Therefore, the status indication of a single IBV may or may not be valid, depending on the status of previous equipments. For this reason, the individual IBV status indication is termed *conditional status.* The function of status resolution is to identify the IBV which truly failed when similar indications are presented from several IBV's simultaneously.

Design Status Reporting and Display BLOCK G.0

The results of status resolution are indications of true status for each IBV in the system. The purpose of Block G.0 is to provide the necessary handling, interface and display of status. This often constitutes signal conditioning and routing to achieve compatibility with interfacing drivers, indicators and/or CRT displays. If the status results are to be used for automatic reconfiguration of the system; drivers, shapers, etc., will be required to effect switching.

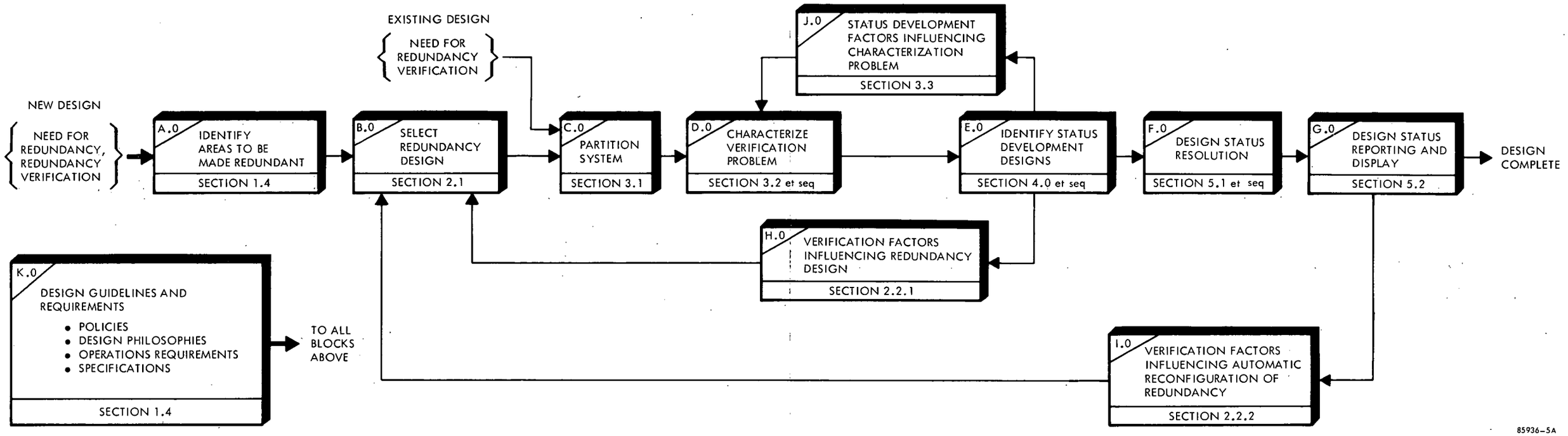


Figure 1.4-1. Design Process Overview

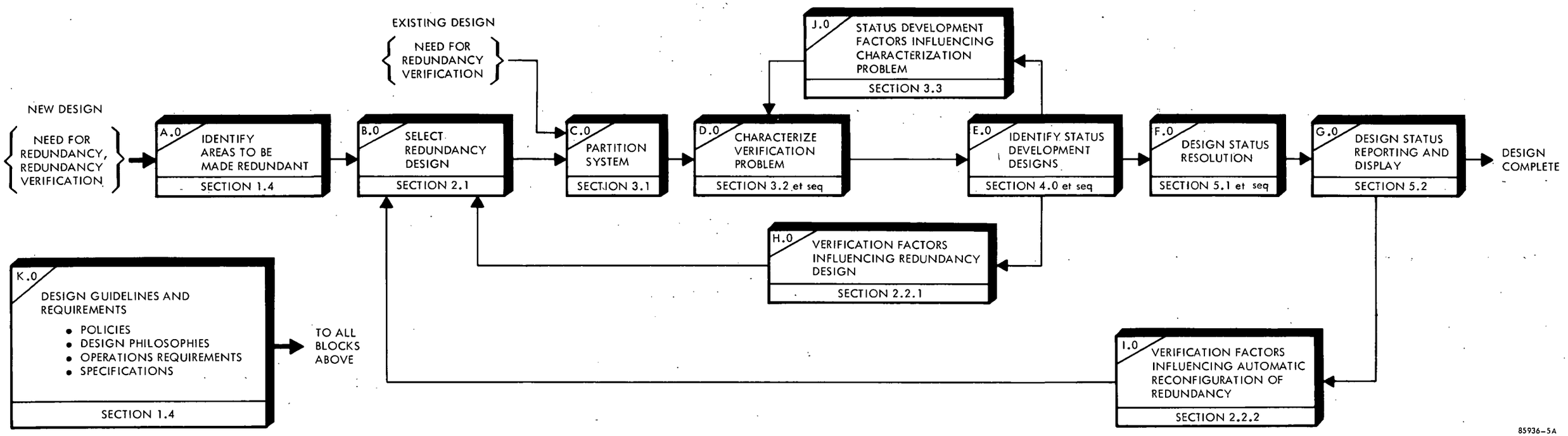
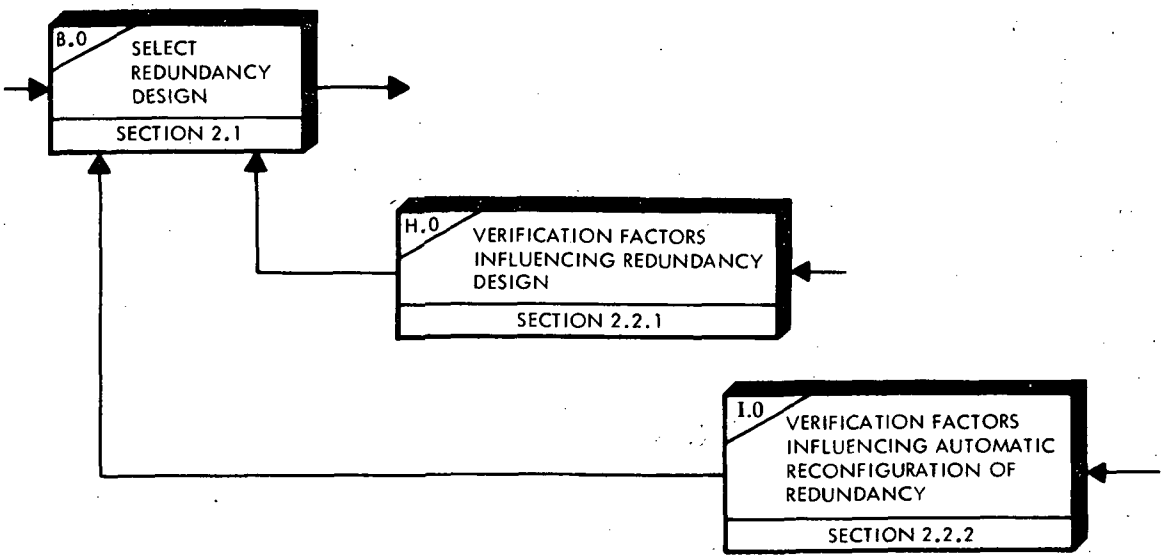


Figure 1.4-1. Design Process Overview
(Repeated From Previous Page)

SECTION 2.0

REDUNDANCY DESIGN



2.0 REDUNDANCY DESIGN

As indicated in Section 1.0, to establish continuity between redundancy and the verification of redundancy, it is essential that a brief treatment be given to redundancy *per se*. Even more significant, however, is the influence which a verification requirement has on the redundancy design. There are some redundancy designs which cannot be verified on an automated basis under present state-of-the-art. The redundancy designer should then be aware of the added constraints imposed on redundancy by virtue of the subsequent requirement of verification. In short, there is little need in achieving an outstanding redundancy design if it cannot be verified.

This section is directed at the redundancy designer to assist him in integrating the needs of redundancy verification into his design. If these needs are understood and considered, the additional time and effort will have been well invested by shortstopping possible major design recycles. The section has been divided into two major areas in fulfillment of this purpose. Section 2.1 addresses the conventional redundancy design methodology and is an expansion of Block B.0 in Figure 1.4–1. Section 2.2 addresses the added impact of verification requirements and is an expansion of Blocks H.0 and I.0 in Figure 1.4–1.

2.1 CONVENTIONAL DESIGN — BLOCK B.0

The conventional approach to redundancy design is being covered solely to provide continuity in the overall design process. As such, the treatment is intended to be brief. The reader wishing more detailed information should consult Reference [1]. Some typical examples portraying the application of the methodology are contained in Section 7.0.

The redundancy design process is illustrated in Figure 2.1 which depicts the next step in detail from Block B.0 in Figure 1.4–1. As such, it is assumed that instances of redundancy have already been identified (Block A.0 of Figure 1.4–1) and the material in this section will be applied to each individual instance. Throughout the handbook, these instances of redundancy are called *redundancy networks* and the entities comprising the networks are called *elements*.

The design process begins by identifying the maintenance approach (Block B.1). This is the first decision point and separates the two gross approaches to the problem. A redundancy approach is considered to be maintained if there is a possibility that a failed element can be restored before all elements in a network have failed. If this is not the case, i.e., if elements can only progressively fail until all have failed, the process passes to Block B.2 and a second decision point. Here the decision of whether to employ switched or unswitched redundancy is made. The term "switched" is used quite liberally to mean any intervening piece of equipment at the output node of a network intended for isolation. If unswitched redundancy is employed, the process passes to Block B.3. Here the design is finalized by identifying the total number of elements in the network and the minimum quantity required to constitute successful operation of the network. Also included in Block B.3 are design considerations for short and open failure modes when piece parts are being considered or for failure independence in more complex elements.

Returning to Block B.2, if switched redundancy is to be used, this block is exited on the remaining path and enters Block B.4. Here the decision is made whether to employ

active redundancy (all elements experiencing operating stress), standby redundancy (off-line elements under reduced or no stress) or majority voting. The design is also finalized in this block.

The remaining block, Block B.5, identifies the decisions to be made if maintained redundancy is to be employed. Of concern is whether maintenance will be effected immediately following a failure or whether a periodic inspection/repair discipline will be followed.

The flow diagram in Figure 2.1 represents a simplified approach and there will be redundancy design situations which will not fit into the neat slots identified. For instance, the concepts of single and multiple path redundancy also apply to maintained redundancy. The approach fulfills basic requirements and the reader should be aware of special cases.

2.2 IMPACT OF VERIFICATION REQUIREMENTS

If a system design is to transition smoothly from an established redundancy approach into the verification design, the redundancy design should consider the requirements of verification. This section identifies two major areas of impact. These are the factors which arise from the immediate need for verification and those which arise from a possible subsequent requirement of using the verification results to automatically reconfigure a system in the event of an element failure. These are treated in Sections 2.2.1 and 2.2.2 respectively.

2.2.1 Redundancy Design Factors

BLOCK H.0

There are four verification factors which directly influence redundancy design of the principal system.

a. The potential exists for a decrease in principal system reliability due to failures induced by physical association with the verification equipment.

b. The fact that verification equipment has been added to the principal system to determine operational integrity, also introduces the risk that this same equipment will inadvertently identify an otherwise acceptable Item Being Verified (IBV) as being unacceptable. This increases the likelihood of failure indications (by adding erroneous indications to existing genuine indications) and can be thought of as effectively decreasing the observable reliability of the principal system.

c. For a given principal system, the greater the number of IBV's, the lower will be the observable reliability due to erroneously identifying an acceptable IBV as being unacceptable. This statement is a direct consequence of (b) above. It says in effect that, increasing the number of IBV's increases the number of decisions to be made. This in turn increases the likelihood of error.

d. For electronic equipment configured in what is often called "parallel" redundancy (i.e., a redundant set), if either

   i) the outputs from the various branches cannot be distinguished, or

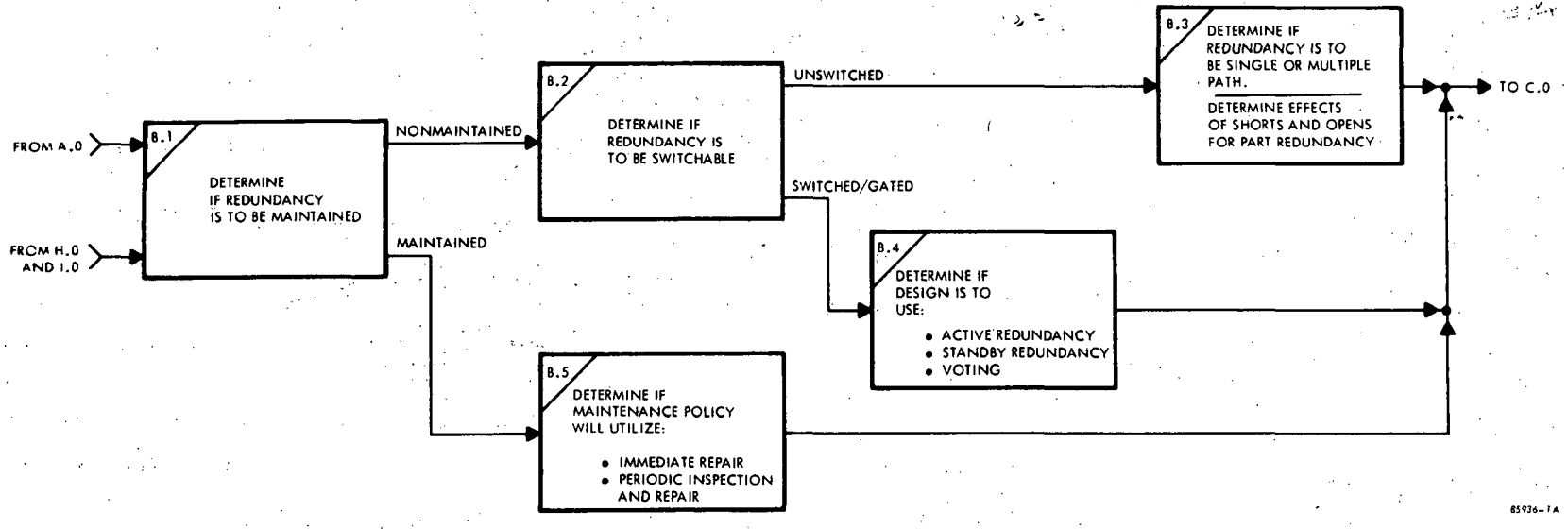


Figure 2.1. Conventional Redundancy Design Process

ii) there is no discernible information at the output node of the network relative to the number of elements operating in the network,

it is quite unlikely that redundancy can be verified at any reasonable confidence.

The verification factors identified in (a) and (b) relate directly to the reliability of an IBV as it is influenced by the addition of verification equipment. Adding verification equipment cannot increase the reliability of a principal system*. Allowances, by way of design margins and apportionments, must be made early in the design if reliability specifications are to be achieved. The extent of the impact on IBV reliability depends in part on the relative complexities of the IBV and its associated verification equipment. This may be more easily seen by examining the sources of the problem more closely. Induced failures, as the term implies, result from anomolies in the verification equipment which "couple" into the principal system through the interface. Induced failures may range from the catastrophic failure of a piece part (which was directly caused by a primary failure in the verification equipment) to an intolerable impedance mismatch at the IBV output. There are two rules for minimizing these effects.

- Utilize a simple isolation device at the interface.
- Keep verification equipment reliability much greater than IBV reliability.

It is seldom possible to achieve perfect isolation and at the same time meet all other design requirements – high reliability being just one such requirement. Also, induced failures can result from reflection back through several stages of the verification equipment.

It is interesting to note that isolation problems are predominantly associated with electronic IBV's. Verification of mechanical, hydraulic, pneumatic, etc., IBV's will almost universally involve transducers to convert the physical effects into electrical signals. These devices provide excellent isolation since they are often "detached" from the process under observation and are very rarely bilateral devices, i.e., anomalies appearing in subsequent electronics will not couple back through to produce alterations in flow rate, rpm, pressure, thrust, etc.

Reduction in observable reliability due to erroneous indications by verification equipment hinges on the premise that a status indication of unacceptable operation will initiate a mission interruption – regardless of cause. Anything that causes interruption of a mission can be construed as nothing other than a failure. These erroneous indications are due to two distinct causes,

- failure of piece parts within the verification equipment, and
- statistical errors resulting from the decision process inherent to verification.

---

*While verification cannot increase reliability, it will supply information about the status of redundancy and thus increase the user's knowledge of system operational capability.

The former cause is directly related to the complexity of the verification equipment, whereas the latter is a direct outgrowth of the design approach and is discussed in detail in Section 4.0. When considering verification equipment failures and induced failures, it can be seen that if verification equipment reliability is high (say one order of magnitude) with respect to that of the IBV, the relative contribution of verification-related failures will be small. It is unlikely that a single instance of verification, being performed on a real time basis, will have a failure rate less than 8×10^{-6} failures per hour. *As a rule of thumb, the practicality of verifying IBV's with failure rates less than 70×10^{-6} failures per hour should be carefully considered.* For comparative purposes, this would typify failure rates for large PC cards, well regulated rack-mounted power supplies, or reasonably complex decoders implemented with Integrated Circuits (IC's).

From the list of verification factors influencing redundancy design presented at the outset of this section, factor (c) deals directly with the level of redundancy employed. This factor states that, due to the statistical errors inherent to the verification process (as identified in the previous paragraph), increasing the number of IBV's increases the number of decisions and consequently the likelihood of error. The implications of this factor directly align with those of the preceding two factors. That is, very small IBV's should be avoided since the division of a system into many (very small) pieces will result in a large probability of error (low observed reliability).

The relationship between an IBV and a single instance of redundancy (a redundancy network) is quite involved and is described in Section 3.0. *For purposes of planning, it is not unreasonable to assume that the number of IBV's will not be less than the number of redundancy networks or greater than the total number of elements constituting these networks**. When applying this rule, the question of redundancy within a redundant network naturally arises. Reference [1] shows this to be less cost effective than using the same material to provide an additional element, thus increasing the degree of redundancy of the network. Therefore, such applications should be limited. It may then be stated that, typically, employing low levels of redundancy infers many redundancy networks; which in turn, result in many IBV's. This assumes, of course, that all instances of redundancy are to be verified. Even if this is not the case, status resolution will usually require status information on intervening equipments (which may require no status information otherwise) to resolve status of a specified IBV.

Turning to the final item in the list of verification factors, factor (d) relates directly to the redundancy approach. Whether verification is possible or not depends first on what type of information is desired about a redundancy network and second, on the approach taken in the design of that network. Verification design considerations regarding this factor are extensive and occupy a good portion of Section 3.0. Suffice it to say that if one is concerned about a single redundant set (see Glossary) the information in Table 2.1 should be sufficient for planning. From this table, the significant factor is obviously distinguishability of outputs. The verification equipment cannot ascertain which element in the

---

*At this stage of the design, these bounds on the number of IBV's are as precise as one can usually be. While the upper bound is firm, the lower bound is an approximation intended to fulfill most situations. It will be shown in Section 3.0 that this bound, under certain verification designs, can be a single IBV, regardless of the number of redundancy networks.

redundant set is at fault unless it can distinguish one from another. A good example of nondistinguishable outputs is the single output formed by hardwiring two electronic amplifiers in parallel. Outputs of the individual amplifiers will form a common node at the junction, and voltage measurements (which are by far the most widely used) cannot distinguish one output from another. It should be noted, however, that whether an output is distinguishable or not oftentimes depends on the sensors available and the implementation of the IBV. The outputs of two lead screws driving a common carriage could possibly be distinguished by strain gauges.

Table 2.1. Feasibility of Verification

Redundancy Design has:	Then Verification is:
1. Outputs distinguishable and common output varies as a function of the number of operational elements.	Always possible
2. Outputs distinguishable and common output does not vary.	Always possible
3. Outputs not distinguishable and common output varies	Possible under centain status information requirements.
4. Neither distinguishable outputs nor common output variation.	Not possible without special measures taken from within elements.

2.2.2 Automatic Reconfiguration Factors

BLOCK I.0

Automatic reconfiguration may be a requirement of a redundancy design. This is particularly true if the principal system will require a policy of immediate maintenance following a failure or if redundant elements are isolated by switching. In effect, the redundancy approach requires status indications produced by verification to complete the implementation of redundancy. A natural consequence of automatic reconfiguration requirements is the necessity of identifying status of each specific element in a redundancy network, i.e., isolation of an unacceptable situation must be to the individual element. Also, there is little need in isolating within an element for purposes of reconfiguration. While this provides added information, there is no mechanism for acting on the information, thus adding cost and increasing the probability of error. Then, with respect to verification, redundancy designs are best implemented as redundant sets and there is little value in being concerned with verification requirements within the elements of the set.

Redundancy verification requires time. Reliable decisions cannot be made instantaneously. If automatic reconfiguration is to be used, it must be recognized that there will be a brief period of time after failure of an on-line element before an operative element can be switched in. This period of time can range from microseconds to several minutes in the worst case. The reconfiguration time results from two sources:

- delay in determining conditional status of a specific IBV, and

- time required to resolve or deduce which IBV in a series chain is actually at fault.

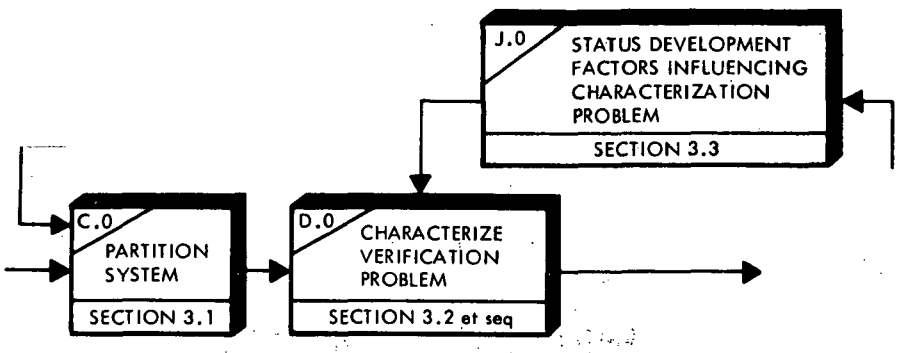
Effects of the latter source can be minimized by several verification schemes if this becomes a significant problem. Delay due to status decisions on a single IBV is typically influenced by other factors and is usually not as amenable to reduction.

The redundancy designer should be aware of verification delay (and its sources) and must specify a response time for automatic reconfiguration.

SECTION 3.0

IDENTIFYING AND DEFINING

ITEMS TO BE VERIFIED



3.0 IDENTIFYING AND DEFINING ITEMS TO BE VERIFIED

As has been indicated in Figure 1.4–1, design of automated redundancy verification begins, obviously enough, with an established redundancy design. This may be an existing design to which verification is to be added or a continuation of a total design effort for a system under development. This section of the handbook is devoted to providing an approach for determining, from an overall system functional diagram, the specific points at which verification will take place, the variables which enter into the determination, the interrelationships between these variables and the significance of each verification point.*

The verification process can be viewed as consisting of two distinct levels:

- design associated with each Item Being Verified (IBV) and which can be considered independent of all other IBV's,
- design associated with the interrelationships between IBV's and circumstances common to all IBV's.

This section provides a means of sorting the given information (from Blocks B.0 and K.0 in Figure 1.4–1) into these two areas and thus defining the problems for each IBV as well as common problems. Once the problems are defined, designing automated verification for each IBV may begin. This is described in Section 4.0. Design common to all IBV's includes interface consideration, status resolution and status reporting. Interface considerations are addressed in Section 3.2.10 with the remaining two areas being covered in Section 5.0.

Two of the most important factors in a verification design are the frequency at which status is to be indicated (i.e., how often) and the approach used to identify status of a particular equipment in a collection of equipments constituting an IBV.** The frequency of status indications determines how often IBV status is examined and should not be confused with a delay in indicating a status change. Any single examination of status will result in a delay between the time the examination is initiated and the time status is actually identified.

It will be convenient to think of the frequency of status indications as being divided into two distinct categories: real time verification and deferred time verification. The meaning of real time verification should be obvious. Deferred time verification accounts for verification performed at set points in time, the interval between these points being large compared to IBV real time operation; e.g., twelve hours. A verification performed only once in the life of a system would also be considered deferred time verification. The trade-off between real time and deferred time verification will involve almost all other factors in a verification design.

Identification of status for a specific equipment in an IBV is the remaining important factor to be discussed. This factor deserves attention since the number of points at which verification directly determines principal system status are not necessarily the

---

*Section 7.0 contains examples applying the material in this section. The reader may find it helpful to periodically refer to the examples.

**Note that all identifiable equipments are not necessarily IBV's. An IBV is followed by a point of verification and has its status indicated with the use of verification equipment.

same as the number of equipments requiring status identification. For example, consider a series chain of equipments and assume that, on an automated basis, status indications are only provided at the output of this chain. There are approaches which can be taken to identify which equipment in the chain actually causes a status change at the end of the chain. This identification is a form of deduction and is discussed in Section 3.2.7. Obviously, the number of equipments in the chain and the method used for deduction determine the complexity of the problem and time delay to an identification. The deduction problem can be simplified by introducing more verification points in the chain until, finally, each equipment has its own verification point. This is known a *exhaustive deduction.*

Deduction and status resolution should not be confused. *Deduction is a process of identifying an equipment in a collection of equipments, all of which depend on a single verification point, i.e., an IBV. Status resolution is a process which identifies, from a collection of verification points, which point reflects the proper status.* Thus, a principal system might consist of several points of verification, each of which is associated with several equipments. Status resolution must be accomplished before deduction.

Referring to the top level flow chart in Figure 1.4–1, this section addresses Blocks C.0, D.0 and J.0. Block C.0 forms the basis for all future design effort and is discussed in Section 3.1. Block D.0 characterizes the verification problem and is considerably more involved than Block C.0. The block requires 14 subblocks to complete the process and is described in Section 3.2. Section 3.3 is devoted to Block J.0 and the feedback of status development factors which may impact the characterization of verification and possibly partitioning.

3.1 SYSTEM PARTITIONING

BLOCK C.0

System partitioning is the first step in defining redundancy verification. The process utilizes the functional flow diagram of the principal system and accomplishes four things.

- It breaks the principal system into manageable pieces.
- It identifies points for verification.
- It identifies potential IBV's.
- It identifies signals available at points of verification.

Simply stated, partitioning amounts to blocking out areas on the principal system functional diagram which constitute identifiable regions for verification. This must, of course, be accomplished by a set of rules. Before identifying these rules, however, it will be necessary to introduce the concepts of *simple sets* and *groups* since partitioning rules and subsequent verification approaches are significantly affected by these concepts.

A simple set is a redundant network constituting a single, independent instance of redundancy. Members of the set (called elements) all have the same predecessors and followers in a functional flow sense, i.e., they all have a common input node and a common output node. Note that a simple set cannot contain other simple sets. No restriction is made as to the complexity of the elements. An example of a simple set is shown in Figure 3.1–1(a).

For purposes of brevity, the term "set" will be used throughout the handbook to describe this configuration.

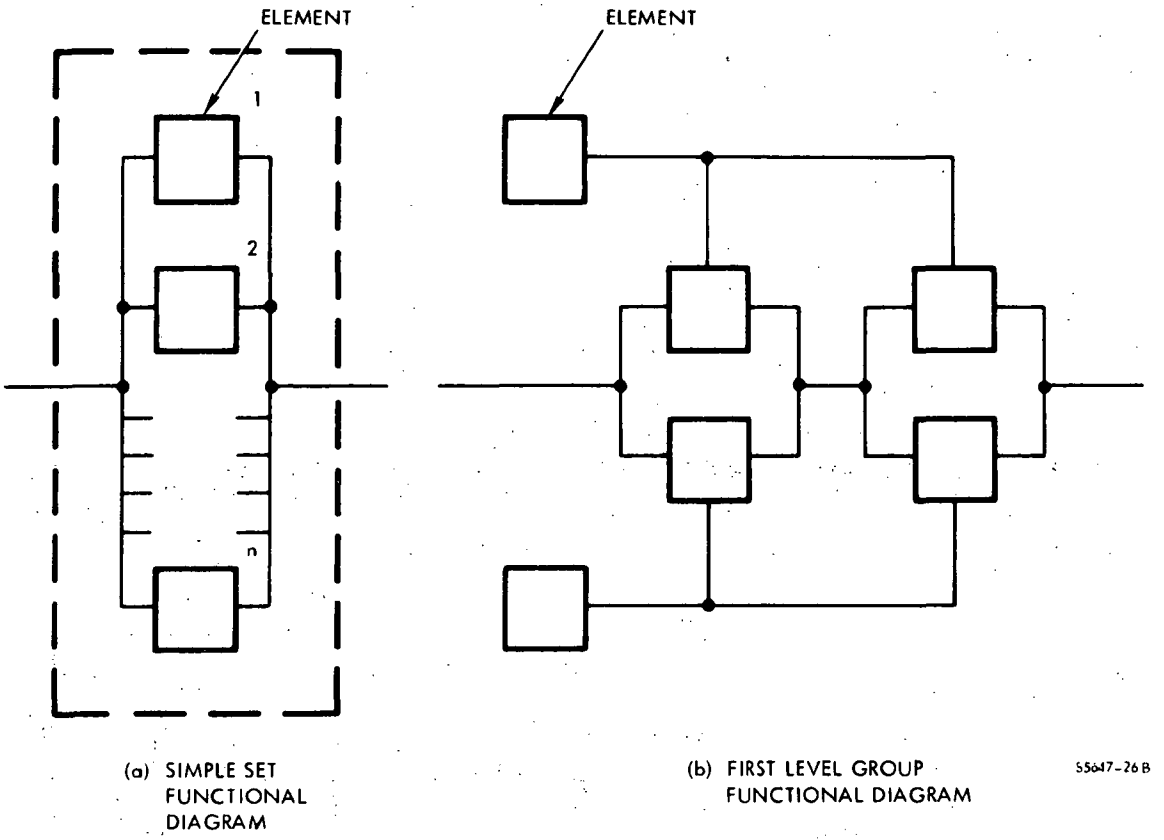


Figure 3.1—1. Examples of Sets and Groups

A group is essentially any arrangement of elements that is not a set. By this definition, a group can contain other groups or more than one set. As a result, it will be convenient to describe groups by levels. A first level group is a group which contains no sets or cannot be further decomposed without loss of functional interrelationships. An example of a first level group is shown in Figure 3.1 1(b). The network is obviously not a simple set since the two pairs of elements, which might otherwise be sets, are fed by a common source (possibly a power supply). Further, subdividing the network would result in loss of functional relatedness. The idea of a group is not the easiest to grasp when seen for the first time. Further examples will be given in a later paragraph.

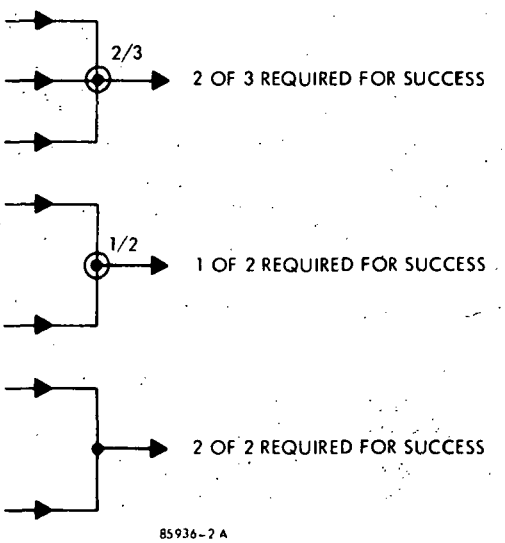
Resuming the discussion of partitioning, recall that system partitioning should be performed on the system functional diagram. The functional diagram must be at a level low enough to identify all redundancy of interest. This does not imply that redundant piece parts routinely appear, however, since Section 2.2.1 indicates it will not usually be practical

to verify these items. Sophisticated parts such as klystron power amplifiers or critical parts such as selected life support controls may be identified. At this stage of design development, the level of detail is not always obvious and *a rule of thumb is to keep the functional diagram at a level above that approximated by a small PCM telemeter, a single coordinate of a stable platform, or rack-mounted oscilloscope.* If, after further investigation, these areas prove to be amenable to further separation and definition, this can always be done. Level of verification is discussed in Section 3.2.1 and it should be necessary here to indicate only that the detail to which verification can practically be applied depends heavily on whether real time or deferred time verification is used.

The partitioning procedure consists of four steps and each is identified below.

Step 1 – Identify all Redundancy

Identify each instance of redundancy and indicate the associated success criterion, i.e., the minimum number of paths required for that instance to be considered operational. The following scheme used at the output nodes will be convenient.



If switches are used at the output nodes, the fractions alone should be satisfactory. It is not always easy to identify redundancy from actual parallel paths on a functional diagram. The above scheme solves this handily.

Step 2 – Identify all Simple Sets

Block off each simple set in the system. This identifies all candidates for exhaustive deduction and instances where an option may exist between identifying status of each element in the set or identifying simply the number of elements which are operative in the set. The latter amounts to indicating how much redundancy is present in the set. It is distinguished from identifying status of each element in that it does not differentiate between elements; it indicates only the quantity. For this reason, the

two methods are described as indicating *element status* or *set status*. Sets identified as requiring verification by themselves, i.e., not included within a verified higher level group, are called *stand-alone sets*. This step will, then, also identify candidates for stand-alone sets.

Step 3 – Identify all First Level Groups

Block off each first level group in the system. This will identify the lowest level areas requiring special attention from the standpoint of determining a deduction scheme. The outputs of these groups will also indicate the minimum number of verification points required.

Step 4 – Identify all Higher Level Groups

Block off each higher level group in the system recalling that higher level groups always contain sets or other groups. There is no theoretical limit on the number of levels which may be used. In practice, however, the number of levels should seldom exceed three. Higher level groups will indicate the approach to status resolution since status at their outputs represent the minimum number of status indications to be resolved.

An example of a partitioned system is shown in Figure 3.1–2. In the figure, Function 1.4 is identified as a group since a single subfunction is feeding what would otherwise be two sets. Function 1.5 is a set since it represents redundancy of degree zero. Since one leg of the parallel paths in Function 1.6 contains redundancy, this collection of elements is identified as a group (a set cannot contain another set). Function 3.4 is a set since point "A" feeds both elements and no other instance of redundancy. Function 3.6 is a set of degree zero redundancy since all three parallel subfunctions are required.

3.2 CHARACTERIZATION OF VERIFICATION PROBLEMS BLOCK D.0

With groups and sets identified as a result of partitioning, the designer can begin selecting the items to be verified and definitizing design requirements for these items. The process is depicted in Figure 3.2*. Steps in the process are summarized below.

a. Identify groups and sets to be verified

b. Determine time aspects of verification for each group/set

c. Identify verification points and any signal injection points for deferred time verification

d. Refine group/set identification and identify all IBV's

e. Definitize requirements for each IBV

f. Identify interfaces

*This figure is included on Page 51 at the end of Section 3.0, *et seq*, for the convenience of the reader.

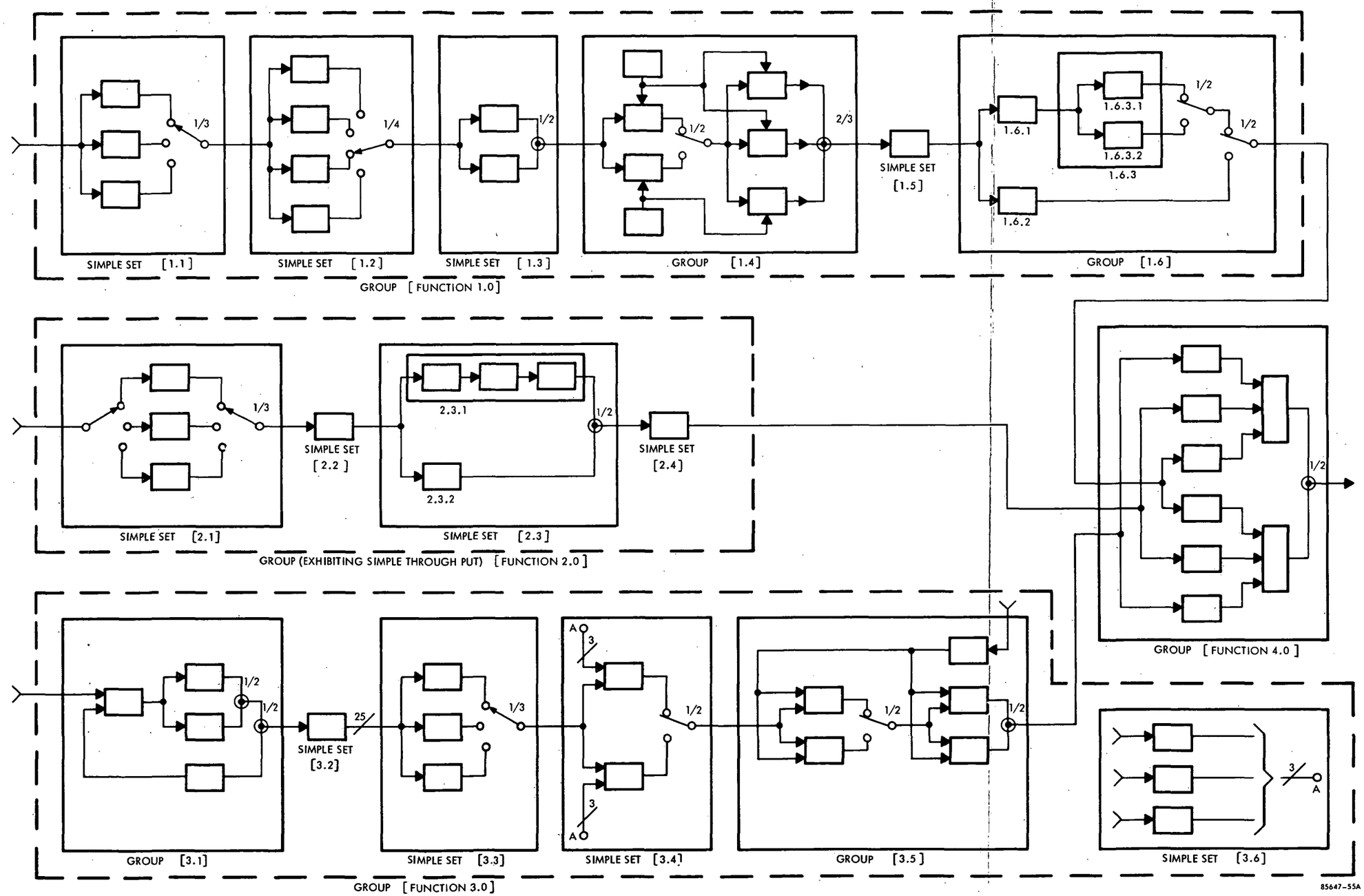


Figure 3.1–2. Sample System Showing Identification of Groups and Sets

As might be evidenced from the summary above, the purpose of this stage in the design is to translate the design guidelines and requirements of Block K.0 (Figure 1.4–1) into requirements for each IBV and integrate these requirements with specific verification constraints.

Before continuing, it will be necessary to point out an underlying premise upon which the remainder of the material in the handbook is based. It has been indicated earlier that status is an indication of IBV operational integrity. The methodology presented herein is based on the symptomatic approach for determining status. That is, the output of an IBV is considered to be its sole function and judgments about operation of the IBV are based on characteristics of the output. Concern is not necessarily with the intricacies of the IBV, but with the quality of its output. If the output of an IBV is satisfactory (in accordance with some frame of reference) the IBV must be operating satisfactorily.

3.2.1 Establishing Level of Verification

BLOCK D.1

Some of the considerations of establishing a level of verification were indicated in Section 3.1. This section will expand the results.

Establishing the level of verification essentially identifies the block of equipment for which status is determined. This block can theoretically range in size from a single piece part to an entire spacecraft. *The lower bound on verification level will be the level of redundancy employed* since, to verify redundancy requires only that each path in the redundant network be verified. There is little need to verify items within a path as this provides superfluous information and can only increase the likelihood of verification error. Function 2.3.1 in Figure 3.1–2 (of the previous section) is a good example. All that is required for redundancy verification is an indication of status for the three items collectively. Indicating status of each adds no more information about whether redundancy is present or not. It is sometimes argued that the added information can be used as isolation indications for maintenance. When considering the cost of adding automated verification equipment and the increased likelihood of error, it is a rare situation which justifies this position. *In general, failure isolation below a redundant path is best accomplished on a manual basis.*

Whether the level of verification should be designed to the lower bound, i.e., to the same level as redundancy, depends on:

- the reliability of the redundant path,
- whether real time or deferred time verification is to be used.

A rule of thumb indicating a practical lower bound for real time verification was cited in Section 2.2.1: it is usually not practical to employ real time verification for an item which has a failure rate less than 70×10^{-6} failures per hour. This rules out the majority of electronic parts and individual circuits. This bound, however, represents only part of the problem. Errors in status indications also contribute. If a system is small and consists of, say, four items having a failure rate around 70×10^{-6}, there will be little concern over the cumulative effects of errors due to the number of verification points. The reliability bound would then be a good guide. On the other hand, a system consisting of, say, 100 items with this

failure rate would present quite a different picture. If each item were individually verified, there would be 100 verification points, each with a fixed probability of committing an error. It would be wise under these circumstances to "double up" on the number of items verified such that only 20 to 50 points would be used.

If time and resources permit, one can conceptually verify redundancy, using deferred time verification and techniques, along the lines of those in existing Automatic Checkout Equipment (ACE), at any level. Such an approach, however, will usually require an interruption of the principal system mission and could be quite time consuming. In addition, each Item Being Verified (IBV) would require access by the verification system and for very large numbers of IBV's this could quickly result in a hopeless tangle of cables, connectors and isolation devices. On the plus side of the ledger is that deferred time verification, (a) will usually not encounter the error problem for larger numbers of IBV's quite as quickly as real time and, (b) can be designed for verifying at virtually any level when the quantity of IBV's is not too great.

In summary, if the level of redundancy goes down to the part level, it is usually not prudent to allow real time verification levels to become this small. Deferred time verification levels might be implemented at part level provided the number of verifications is "reasonable."

Collections of hardware constituting the lowest level of interest to verification will be called elements. If the level of redundancy is below the level of verification, these elements may contain redundancy. This redundancy is obviously not verified.

The upper level of verification can theoretically range to a single verification for an entire principal system. If the principal system consists exclusively of sets, this can be a practical approach. Under these conditions, the entire system would constitute a second level group and a deduction scheme can be used to determine the element at fault when status of the group changes. If the system consists of first level groups, the picture changes somewhat. First level groups require a deduction scheme of their own, and utilizing deduction schemes inside deduction schemes can cause ambiguities to arise. As such, first level groups will require a verification point.

In summary, when considering the upper level of verification, each first level group will constitute an instance of verification, i.e., status will be provided for the group. Contiguous (in a functional flow sense) sets may be identified as a group and constitute a single instance of verification.

With bounds established, the problem of identifying groups and sets to be verified is now manageable. It should be remembered, however, that the groups and sets identified thus far may not constitute all points to be verified. Specifications may require status on each element in a set, especially if the elements differ widely in capability or reliability. Deduction schemes for first level groups may require status on selected elements within the group. These points are discussed in Section 3.2.7.

3.2.2 Real Time Versus Deferred Time Verification

BLOCK D.2

Having identified the groups and sets to be verified, requirements for each must now be identified. The first requirement to be identified is whether the group/set is to employ real time or deferred time verification. *The five most important aspects influencing*

this decision are the importance of the duration of a failed condition, the criticality of a failure, the inability to interrupt principal system mission, the accessibility of the group/set to be verified, and the maintenance policy. Real time verification will likely be required if:

- failure of the group/set is critical,
- unidentified false information is important (e.g., command destruct transmission),
- principal system mission cannot be interrupted, or
- maintenance is to be effected immediately after failure.

On the other hand, if:

- the group/set is not accessible for real time verification (e.g., an aircraft during flight with no verification equipment aboard),
- real time verification is not necessary to accomplish the verification goal,
- real time verification will result in an unacceptably low confidence in status indications,

then deferred time verification will likely be the candidate.

Recall that deferred time verification can range in frequency of verification from once during the life of the principal system to rates approaching real time. Once deferred time verification is elected, a verification rate must be selected for each group/set. There is no requirement that the rates for all groups/sets in a system be the same. Each will be selected on the basis of individual circumstances and principal system features common to all groups/sets, e.g., mission profiles, operations requirements and mission criticality.

One application of high rate, deferred time verification is sampling. If there are several identical IBV's, it may be cost effective to time share verification equipment. This approach has the potential of offering considerable savings over dedicating verification equipment to each IBV.

It was indicated above that real time verification would likely be required if the principal system mission cannot be interrupted. This point deserves further clarification. Real time verification, by its definition, will always perform verification without interrupting or interfering with the principal system mission. Depending on the circumstances, deferred time verification may or may not require mission interruption. If:

- there exists no "natural gaps" in the mission profile of the group/set under contention such that verification can be performed during these periods, or
- verification cannot be adequately performed without injecting signals into the principal system which would preclude that system from performing its mission,

then deferred time verification will require interruption of the mission. Otherwise, deferred time verification can be performed without mission interruption and is achieved quite similarly to real time verification.

In summary, to achieve real time verification, one must be able to implement the design without interrupting the principal system mission. Deferred time verification need not interrupt the mission. However, one prominent reason for shifting from the desire for real time verification to using deferred time, is that insufficient real time information can be achieved. Under these circumstances, mission interruption is the only alternative. The above reason is related to confidence in status indications. This is the subject of Section 3.2.4.

A final point to be made in this section regards a distinction in the use of the term real time verification. Consider a set consisting of a redundant pair or equipments. Further assume that status of each equipment is to be determined, i.e., each is an IBV. Referring to real or deferred time verification for the single IBV is clear enough: note, however, that real time verification of one IBV and deferred time for the other does not constitute real time verification of *redundancy*. Care should be taken to place in proper perspective, *what* is being verified real time. If redundancy is to be verified real time and status of all elements in the group/set is required, each must be verified real time.

There is a special interpretation of the preceding statement which bears considerable practical merit. Consider again, the redundant pair of equipments. This time assume that one has a failure rate two orders of magnitude greater than the other. If the high failure rate item is verified real time and the low failure rate item deferred time, it may, for all practical purposes, be assumed that *redundancy* verification is also real time. In effect, redundancy verification results would be the same as those of the high failure rate item.

3.2.3 Time Profiles

BLOCK D.3

Section 3.2.2 indicated that an effective method of performing deferred time verification on a noninterference basis was to exploit existing gaps in mission profiles. This allows the use of techniques which would otherwise disrupt principal system performance. Time profiles will provide the necessary information by indicating the duration (or as a minimum, the statistics of duration) of time available for verification. If time profiles are available for each group/set, and these profiles are registered to a top system profile, it may be possible to find blocks of time when groups/sets are not required during the mission. If the blocks recur in time, their occurrence rate may well establish the verification rate. Some systems, such as the Space Shuttle, will have successive missions interrupted by periods of time on the ground for refueling, servicing and the like. This is a block of time during which all groups/sets are available (or can be made available) for verification. It is quite possible that the time between such blocks will be too great to afford sufficient confidence between verifications. If this is the case, a hybrid scheme involving real and deferred time verification may be necessary.

Time profiles will also provide information for real time verification. Section 3.2.6 describes the importance of alternate IBV modes of operation. If these modes are programmed, they will appear on a time profile. Knowing the occurrence and duration of these modes will afford planning and allow programmed application of the strategies described in Section 3.2.6.

3.2.4 Confidence Levels

BLOCK D.4

Confidence expresses one's willingness to believe. The implications of confidence appeared in preceding sections as a recurring theme. This is not coincidental. Probably the most far-reaching and difficult task of a redundancy verification design is that of establishing a desired confidence in the resulting status indication. It is one thing to say that the results should always accurately represent the true picture and quite another to achieve this end. It is a foregone conclusion that a confidence of 100 percent cannot be realized. If signals are measured, errors are introduced. If they are compared in some manner, one must ascertain what degree of similarity is desired. If threshold techniques are used, the decisions are subject to error. The designer must settle for something less than 100 percent confidence. It is interesting to note that, if no verification existed, no errors could be made regarding status (one would simply remain ignorant of status). In effect, the problem involves trading off information about status (decreasing one's uncertainty of this subject) with the fact that some of this information may be incorrect and falsely influence operations decisions. It is important to ask the worth of this trade-off. How much is it worth not to blindly assure redundancy is present and find out ultimately, perhaps catastrophically, that it actually was not present? Is it worth the delay and cost of an indication that redundancy was not present when indeed it was? The verification designer must have some notion of the desired confidence, for this factor enters into virtually every trade-off.

Confidence in a status indication involves two distinct areas of consideration: (a) the ability to extract sufficient information to represent operational integrity and (b) once the information is extracted, the susceptibility of verification equipment to errors. The former consideration is one of judgment. How much information is required to convince a designer that the IBV is performing in an acceptable manner? Acceptable operation can take on a range of connotations from requiring compliance to complete specifications, to requiring only the presence of a signal. Interest will typically lie somewhere between these two extremes. Practical considerations frequently limit implementation of high confidence techniques demanding complete specification requirements and the designer must decide on a "reasonable" compromise. It is very difficult to quantify how well a design must faithfully reproduce complete performance indications and sound engineering judgment must be relied upon. *As a rule of thumb, real time verification will seldom yield indications approaching those of complete performance compliance.*

The second consideration of confidence is that of decision error. Once operational information is decided upon, equipment must be designed to make status decisions based on this information. This equipment will make decision errors. What is the effect of this on the total confidence in design? Consider the following hypothetical case. If one is to make five independent decisions and his probability of a correct decision is 0.6 for each decision ($p = 0.4$ of an incorrect single decision), the probability of at least one of the five decisions being incorrect is $1 - (0.6)^5 = 0.92$. The results are quite revealing and a conclusion to be drawn is to simply minimize the number of decisions. Decisions are being made at numerous points and each point consists of a time sequence of decisions. The first place to reduce decisions is to reduce the number of decision points. This typically implies that IBV's should be made as large as possible. Instead of making five decisions on five separate entities, lump the five decisions into a single IBV.* This approach will also improve the results from a

*This decision could radically affect the redundancy design. If the collection of five equipments are not to be treated as a group, the level of redundancy may have to be moved upward to correspond to the verification level.

tolerance viewpoint. It may be that the drift on two entities is marginal but in opposite directions. If a decision is made on each entity, one (or both) could be identified as unsatisfactory. However, if a decision is made on the output of both, the drifts may well cancel leaving the ultimate output satisfactory. And isn't this all that can be asked of equipment? Making decisions on too small a level will tend to make the verification quite sensitive (if not hypersensitive).

The possibility of more than one level of confidence should not be overlooked. It is entirely feasible to require two levels of confidence for a system. One level would critically verify the integrity of operation, approaching compliance to specifications. The other level would verify operation to a "reasonable" or "quick look" confidence. The former level of verification would likely be performed on a deferred time basis, requiring the principal system to interrupt its intended function while verification is performed. The latter could be performed real time on a continuous basis while the principal system is performing its intended function. The former approach would likely require injected signals and predefined checkout sequences typical of most Automatic Checkout Equipment (ACE) today. The latter approach would represent a lower confidence but has obvious advantages. When considering the remainder of the material in this section, it should be recalled that it may well have to be applied to more than one confidence level. Different techniques would likely result from considering the two verification approaches above.

Confidence, then, consists of two areas: representation of performance compliance and verification errors. A designer must have at least a guideline to indicate desired confidence. While representation of performance compliance cannot be quantified, error probability can. Each IBV should have an error probability assigned. These errors consist of two types. One type is committed when the verification equipment identifies an acceptable IBV as being unacceptable. For convenience of notation, this is called a Type I error. The observable reliability of the principal system is affected by Type I errors. As a result, interest in these errors extends throughout the entire mission of the principal system.

The remaining type of error is committed when the verification equipment does not indicate an unacceptable IBV within a prescribed notification time. This type of error is called, logically enough, a Type II error. Interest in Type II errors extends only over the notification interval.

To complete the specification requirements, the following must be identified:

- Probability of a Type I error
- Probability of a Type II error
- Mission Time
- Notification Time

The two probability requirements will usually be budgeted or apportioned, from an overall principal system requirement, to each group/set. Once IBV's have been identified, these apportionments must be adjusted accordingly.

3.2.5 Signal Injection for Deferred Time Groups/Sets

BLOCK D.5

Before addressing signal injection, it will be necessary to distinguish among the several types of signals to be discussed. The signals present in the principal system, before verification is considered, are termed *tenant signals.* The verification design may require special, additional signals to be injected into the principal system. These signals are part of the verification design and are termed *injected signals.* Finally, there will be signals peculiar to the verification equipment. These signals are dedicated to carrying verification information and will be called *verification signals.*

There are three types of injected signals and a clear distinction among these is essential to determine the feasibility of real time verification. Suppose that an IBV is to be verified real time and that attempts at extracting sufficient information from its output signal regarding operational integrity have proven to yield marginal confidence in status results. Must real time verification be abandoned? Perhaps not. If it were possible to devise an injected signal which could somehow be superimposed on the tenant signal (e.g., multiplexed, modulated) and this superimposed signal in no way interfered with the performance of the IBV, this injected signal could supplement existing output verification information. Under these circumstances, real time verification could still be implemented. The conditions describing this situation are too involved to repeat each time a signal meeting these requirements is discussed. It will be convenient to identify an injected signal possessing the properties described above as a *symbiotic signal.*

Suppose that either (a), very high verification confidence was desired or (b), real time verification could still not be accomplished with the use of a symbiotic signal. The only remaining way to accomplish verification is to inject a signal which will accomplish verification, but which will also interfere with the tenant signal and thus alter performance of the IBV. Obviously, such a practice must be performed deferred time. The injected signal must simulate the tenant signal in certain important respects if confidence in status results is to be achieved. The fundamental distinction between the injected and tenant signals is that properties of the injected signal are known at each point in time. An injected signal which fulfills the requirements above is known as a *simulative signal.* Note that simulative signals require interruption of the IBV mission and will typically yield high confidence status results.

There is a third type of injected signal which finds limited use. Consider an IBV which remains essentially passive during its mission and the fraction of time a tenant signal is present is very small compared to the total mission time. Such an IBV might be incorporated in a range safety system or certain other command/control systems. Concern to this point has been with whether or not to use the tenant signal for verification. What if no tenant signal is present? A symbiotic signal does not apply and it is doubtful that a simulative signal can be used since this would increase the likelihood of duplicating the tenant signal. Such system approaches usually involve large portions of idle time and handle a finite repertoire of tenant signals. If such an IBV is to be verified real time, an injected signal must be used which does not match the repertoire of tenant signals, yet provides confidence in operational integrity. Such injected signals are known as *idle signals.* The relationship of signal types is shown in Figure 3.2.5. Injected signal uses can be summarized as:

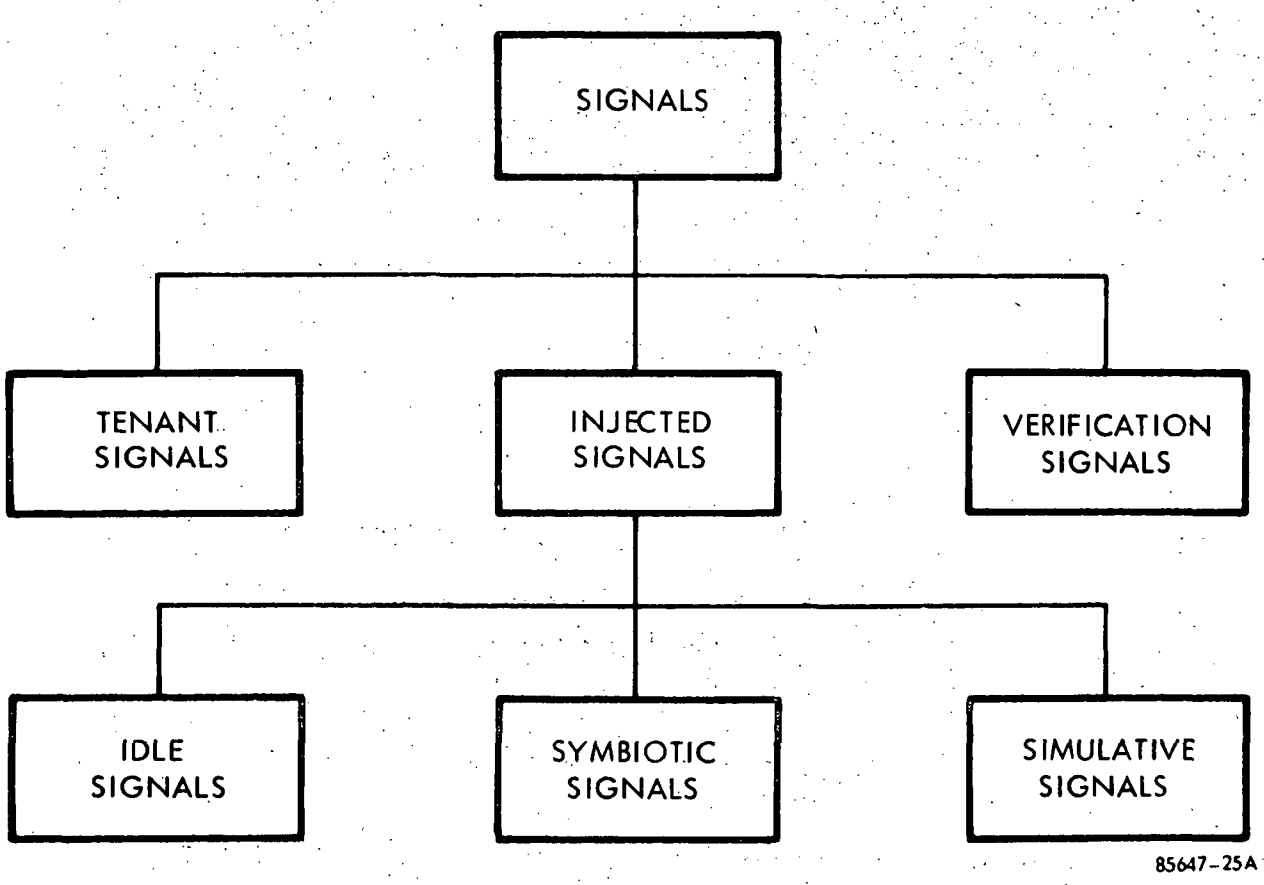


Figure 3.2.5. Signal Relationships

Symbiotic signals – used to augment tenant signals.

Simulative signals – used to take the place of tenant signals.

Idle signals – used to fill large time gaps when tenant signal is not present.

With distinctions being drawn among the several signal types, the remainder of this section will be devoted to the identification of signal injection points in the principal system. The discussion will be limited to simulative and idle signals since identification and selection of symbiotic signals must await design which is developed in Section 4.0.

Based on design requirements identified in Section 3.2.4 and system operations concepts, each group/set can be identified as to whether real or deferred time verification is required. Groups/sets which are to be verified deferred time and which will require high confidence, will typically require injection of simulative signals. The simulative signals provide a known base from which to judge operational integrity of IBV's and each injection point will establish a frame of reference for subsequent groups/sets. Should an unacceptable condition be indicated at a verification point, it will not be necessary to ascertain the status of IBV's prior to an injection point. Status resolution and/or deduction

may thus be limited to segments between injection points. Signal injection points will, then, influence system partitioning and will typically identify new groups. Referring to Figure 3.1–2 on page 29/30, suppose a simulative signal were injected at the input to Function 1.1 and a second injected at the input of Function 1.4. Functions 1.1, 1.2 and 1.3 would constitute a new group, separated from Function 1.4 by the simulative signal injected at its input. Verification must be performed at the output of Function 1.3 as a minimum and deduction schemes may be limited to those three functions. By increasing the number of simulative signal injection points, one may reduce the size of second level groups. If each first level group and set in the system is provided with an injection point at its input and verification point at its output, there will be no need for status resolution. Sets with either a simulative signal injection point or a verification point at their input and verification point(s) at their output are called stand-alone sets. These sets may be treated independently in later portions of the design.

Injection points for groups/sets requiring real time verification using idle signals can be determined from the requirements identified in Section 3.2.4 and from functional descriptions of their design. Typically, idle signals will be injected on the first point (in a functional flow sense) where quiescent operation is indicated.

3.2.6 Other Pertinent IBV Properties

BLOCK D.6

To this point it has been assumed that there would be a causative relationship, through all time, between the output and the operation of an IBV. That is, there is no part of the IBV which is not contributing toward producing the output. This can be violated in two ways: (a) the IBV contains nonlinearities (e.g., limiters, clippers, saturation, etc.) such that regions of the input signal will not effect a response in some portions of the IBV (the IBV is termed as not being exercised); and (b) the IBV contains distinct operating modes such that at least one mode does not require all IBV functions. If either of these two conditions exist, knowing status of the output will not be the same as knowing the status of the IBV. If the time during which the IBV is not being exercised is quite short, little information will be lost and a distinction between status of output and IBV is academic. On the other hand, if the time is known to be long and if the function not being exercised is critical, the distinction can be significant.

Time profiles will usually be the first indication of multimode IBV operation and design descriptions should indicate existence of nonlinearities. These factors must be identified and passed on as part of the IBV verification design requirements. Final identification of IBV's will be made in the next group of blocks (Figure 3.2). However, with the group and set identification existing to this point, tentative IBV definition will be sufficient to fulfill these requirements.

Multimode operation can be accommodated by altering verification parameters as the IBV changes operating modes. This will not, however, account for portions of the IBV which may not be exercised. Probably the surest method of verifying an IBV which is not being exercised by a tenant signal is to inject an exercising signal. This signal will inevitably be a simulative signal and thus require interruption of the mission. Alternatively, signals may sometimes be extracted from within the IBV (in addition to its output) to gain added information regarding its operation. For example, an indication of the AGC voltage in an AM receiver would provide added information on which to base output amplitude measurements.

3.2.7 Specific Group and Set Approaches

At this stage of the design, all groups, regardless of level, and all *initial pass* stand-alone sets should have been determined.* Due to the analysis performed in Blocks D.1 through D.6 (Figure 3.2), groups identified during initial system partitioning should be somewhat modified and better defined. The character of the partitioned system will then be changed to the extent that three distinct and nonoverlapping equipment arrangements remain.

a. Initial pass stand-alone sets

b. First level groups

c. Second level groups which contain neither first level groups nor initial pass stand-alone sets

This has been the object of the design to this point. What has occurred is that some of the initial partition sets have been identified as stand-alone sets. Those that have not, were absorbed into second level groups. Any third level groups, having served their initial purposes of organizing the principal system and providing design alternatives, have been eliminated. All second level groups have been redefined such that they contain no first level groups. Due to signal injection, some initial second level groups (which contained no first level groups) may have been divided into two or more second level groups.

The above rearrangement of partitions** has resulted from more complete information about the design and is essential if all IBV's are to be verified. The current state of IBV definition can be summarized as:

- initial pass stand-alone sets will be IBV's themselves or their elements will be IBV's,
- all first and second level groups are tentatively considered to be IBV's.

This section of the handbook makes final identification of IBV's using the information summarized above and determines group deduction schemes. Approaches to sets and groups are quite different. The section treats each separately with Sections 3.2.7.1 through 3.2.7.3 addressing the stand-alone set problem and Sections 3.2.7.4 and 3.2.7.5 addressing the group problem. It should be noted that the order of treatment is not necessarily the order in which a designer would approach the problem. This is due,

---

*The reason for identifying these stand-alone sets by "initial pass" will become clear in Section 3.2.7.4.

**It is important to note that if a hybrid approach of real time and deferred time verification is employed, it is entirely possible that two distinct partitionings may result.

in part, to the separate treatment of the stand-alone set. A stand-alone set can be identified during initial system partitioning or as a result of identifying exhaustive deduction as a group approach in Block D.10. The actual order of design progress would be the solution of second level groups first, followed by first level groups and sets solved virtually in parallel. It is this flow that Figure 3.2 attempts to depict. No distinction has been made between approaches to first and second level groups since the design process is the same. All references to sets in the succeeding material should be understood to mean stand-alone sets. All references to second level groups should be taken to mean such groups which contain neither first level groups nor initial pass stand-alone sets.

Before considering details of the two design processes, it should be fruitful to summarize some of the differences between groups and sets.

- By their nature, the status of a group (analogous to status of a set) will typically have no meaning, for knowing three of five elements in a group are operative, will seldom tell the desired story. However, it may be possible to attribute a status statement of the form – all elements operative, not all elements operative – to a group.

- Unlike the set where (frequently) one element is on line and the option of using simulative input signals to the remaining elements may exist (see Section 3.2.8), the interconnection of elements in a group will require (usually) that more than one element use the tenant signal.

- In a set, knowing the status of one input (although it may have many independent information paths) will confirm the conditional status of an element. In a group, this will typically not be so.

3.2.7.1 Distinguishable Set Outputs

BLOCK D.7

It has been indicated in Section 2.2.2 that certain redundancy approaches to sets cannot be verified (or require at best, special verification design approaches and extended equipment sophistication). Such approaches cause obvious concern. The first test to determine if a stand-alone set is verifiable is to ascertain whether element outputs can be distinguished. If element outputs can be

- discernibly separated, and

- uniquely identified to the proper element,

then verification is always possible. Separation can be achieved in several ways. The most common of these are:

a. physical separation (mechanical elements,

b. electrical isolation using switches, gates or isolating diodes,

c. time and frequency using multiplexing of element outputs,

d. current sensors for electrical/electronic elements.

(The reader should refer to Section 2.2.2 for a further discussion of distinguishable outputs.) It should be noted that not all schemes designed to achieve failure independence in a set can provide distinguishable outputs. The use of isolating diodes in (b) above usually comes about from attempts to achieve failure independence. When properly exploited, such devices can, however, provide distinguishable outputs for limited purposes.

The use of current sensors deserves additional comment. One of the classic problems of electrical/electronic equipment is the set whose element outputs are connected together to form a common output. (This is known as hardwired output.) With the exception of high frequency phase differences, voltage measurements at the two outputs will indicate the same quantity. Current measurements, if they can be implemented under present state-of-the-art, will usually provide sufficient information to distinguish one output from another.

If the element outputs are distinguishable, consideration may turn immediately to identifying IBV's. If not, there is one additional test to apply and this is the subject of the next section.

3.2.7.2 Variation of Set Output

BLOCK D.8

As indicated in the preceding section, hardwired outputs on electronic equipment present the most significant stumbling block to distinguishable outputs. If the outputs are not distinguishable, there still remains a possible alternative. The common output of a hardwired set will vary somewhat with the number of elements providing proper outputs to the common output node, i.e., as elements fail, characteristics of the common output change. If this change is *predictable* and *discernible*, it can be used to indicate the number of elements operating in a satisfactory manner. The requirement of being predictable is self explanatory and may not be a significant problem in all cases. Discernibility, however, will present a problem in virtually every case. If a set is to be truly redundant, the output cannot vary too much as elements fail. As a result, the changes will usually be small and often not result in a linear relationship to the number of operable elements.

In summary, even through element outputs are not distinguishable, if a common output varies such that it provides an indication of the number of operable elements, redundancy may still be verified. If neither condition exists, redundancy verification is quite likely not possible and consideration should be given to altering the redundancy design.

3.2.7.3 Status of Set vs. Status of Element

BLOCK D.9

There are two ways of indicating status for stand-alone sets. These are:

a. indicate the status of each element in the set, or

b. indicate the number of operable elements in the set, or equivalently, status of the set.

The method in (a) above distinguishes among the elements and (b) does not. It may seem that redundancy verification should only be required to identify how much redundancy is present, not the operational integrity of each element. There are several reasons for requiring status of each element.

- The capability and/or reliability of the several elements may be quite different. This will usually result from principal system recovery policies such as fail soft, fall back, or fail-operational/fail-safe.
- If elements are switched, either manually or automatically, the specific failed element must be identified.
- If maintenance is to be effected immediately following a failure, the specific element which has failed must be identified.

If circumstances require a distinction between status of elements in a set, the status of each element must be determined. Each element is then an IBV and the several outputs must be distinguishable.

Set status may be determined by using either the approach of variation of a common output or the approach of distinguishable outputs. If distinguishable outputs will allow status indications of each element, it will certainly allow the status of a set to be determined. However, if a discernible variation in a common output is achievable, this approach will typically be more cost effective than determining status of each element – especially when it is desired to only indicate set status. If a discernible variation is not achievable, set status may still be achieved if element outputs are distinguishable. Aside from the fact that set status may be all that is desired, there are four other pertinent reasons for requiring set status over element status.

- The packaging may not warrant the additional detail. It may be that the entire set is packaged as a single line replaceable item.
- The cost of identifying the status of each element may be prohibitive.
- The outputs of each element may not be distinguishable. This would make the status identification of each element impossible.
- Any automated scheme for identifying the status of each element may prove to be more unreliable than the set itself.

Upon completion of efforts indicated for this block in Figure 3.2, each set in the system should have a policy statement indicating whether element status or set status is required. This will indicate the IBV's for all sets, i.e., the element is the IBV for element status and the set is the IBV for set status.

3.2.7.4 Group Deduction Schemes

BLOCK D.10

It was noted in Section 3.2.7.3 above that stand-alone sets could be treated on their own merits, resulting in (at least in principle) a comparatively straight forward verification approach. This is largely due to the fact that, in a functional flow sense,

there is but a single element between set input and output. Groups do not necessarily possess this feature. Like the stand-alone set, a group may have a verification point at both input and output (or alternatively, a signal injection point at the input), however, there will be more than one serial element between these points. If status of intervening elements in a group is to be determined, it must be somehow deduced from the group verification results. There are three schemes for deducing status of entities within a group. These are:

- exhaustive deduction,
- iterative deduction, and
- logical deduction.

Practical designs may well use combinations of the latter two.

Exhaustive Deduction

As the name may suggest, exhaustive deduction does not even involve deduction in the usual sense of the word. Instead of trying to rely on group status alone, the scheme requires status to be indicated (not deduced) for each entity in the group. The result is a division of the group (a single, large IBV) into smaller IBV's. For example, if a second level group consisted of four sets in series, exhaustive deduction would divide the group into four new stand-alone sets. (Thus the qualification of "initial pass" for those originally identified.) If a first level group consisted of a mixture of sets and other arbitrary elements (see Figure 3.1–2), each element and set would require verification. It should be recalled that a single element is also considered to be a (degenerate) set.

The use of exhaustive deduction may seem trite until it is compared with the remaining two schemes. Many practical situations can only be resolved by this scheme. In effect, exhaustive deduction amounts to a second partitioning of the group and the material in Sections 3.2.7.1 through 3.2.7.3 should be applied in each case. This is indicated in Figure 3.2 by the path returning to Block D.7. The use of exhaustive deduction over one of the other deduction schemes should have compelling reasons since (as might be anticipated) this scheme will usually be the most costly to implement. In its favor, however, is its inherent compatibility with real time verification requirements of no interference of principal system mission

Iterative Deduction

Iterative deduction implies the ascertaining of group status by performing iterative switching of redundant elements. This can be achieved through two basic approaches which essentially amount to real time verification on the one hand and deferred time verification on the other. Considering the real time verification approach first, the ploy here is to approach verification of on-line and off-line elements in two distinct ways (see Section 3.2.8). The on-line elements are obviously all connected to perform the intended group function and these elements are verified as a collection by observing only the group output on a continuous basis. The off-line elements are configured in whatever fashion proves convenient for their real time verification. So long as the on-line elements are not involved, virtually any partitioning and verification scheme may be used.

The above will accomplish real time verification. However, what will happen when an on-line element fails? How is element status determined? When the on-line verification technique determines that the group on-line string has experienced a failure *somewhere,* it will initiate a search routine which begins substituting off-line elements, one at a time, until the failure indication is removed.* The last switching occurrence, the one that removed the failure indication, is identified with a specific on-line element and that element's status is declared unacceptable. Since the off-line elements are being verified real time, elements of known status are being substituted at all times. Should an off-line element be in a failed state when the switching routine is initiated, the iteration processor is informed of this condition and skips this substitution. If all possible substitutions do not remove the group on-line failure indication, status of the skipped on-line element is declared unacceptable. From this discussion it can be observed that this approach to iterative deduction enjoys its greatest utility when used on maintained principal systems.

The deferred time verification approach differs from that of real time in one important respect: the off-line elements are not independently verified. Verification of the group is still based on the group on-line output, but periodically the group must be released from its operating function while off-line elements are sequentially switched into the on-line stream for verification. By using a technique which will switch in one new element of each set for each output observation (i.e., each new observation constitutes a completely different collection of off-line elements), redundancy verification can be accomplished quite rapidly if no unsatisfactory condition is indicated in the process. For then, the number of system configurations requiring examination will be a minimum. However, if an unsatisfactory output is observed for one or more of the configurations, the only information gained is that some one element which is on-line in that configuration is unacceptable. A switching sequence must then be undertaken to identify the faulty element and, even though an optimum sequence can probably be developed, its location will usually be time consuming.

If the design confidence requirements are such that an input for verification (simulative signal) must be varied over some range of frequency, amplitude, etc., the necessity of doing this in all different system configurations could be a lengthy process. In any event, just reconfiguring the system and allowing the verification equipment to settle after each switching instance is time consuming. This is the greatest drawback to iterative deduction since it will typically be less costly to implement than exhaustive deduction and is not as susceptible to errors.

Logical Deduction

The concept of logical deduction, when applied to redundancy verification, implies the use of knowledge concerning system configuration and signal flow to minimize the number of points which must be observed to deduce status. The goal of the scheme is to derive a maximum amount of status information by intelligently choosing a minimum

---

*It is important to note that this process is based on the principle of single instances of failure. If the iteration time is short, typically within minutes, this assumption is valid.

number of verification points. The scheme is best described with the use of an example. Consider Figure 3.2.7.4 where the input of A effects an output only at A′, B at B′, and C at C′. Suppose an "unsatisfactory" output was observed at A′ while outputs at B′ and C′ appeared to be good. A logical conclusion might be that boxes A or A′ are at fault, since any failure in X or L would have shown up at B′ or C′.

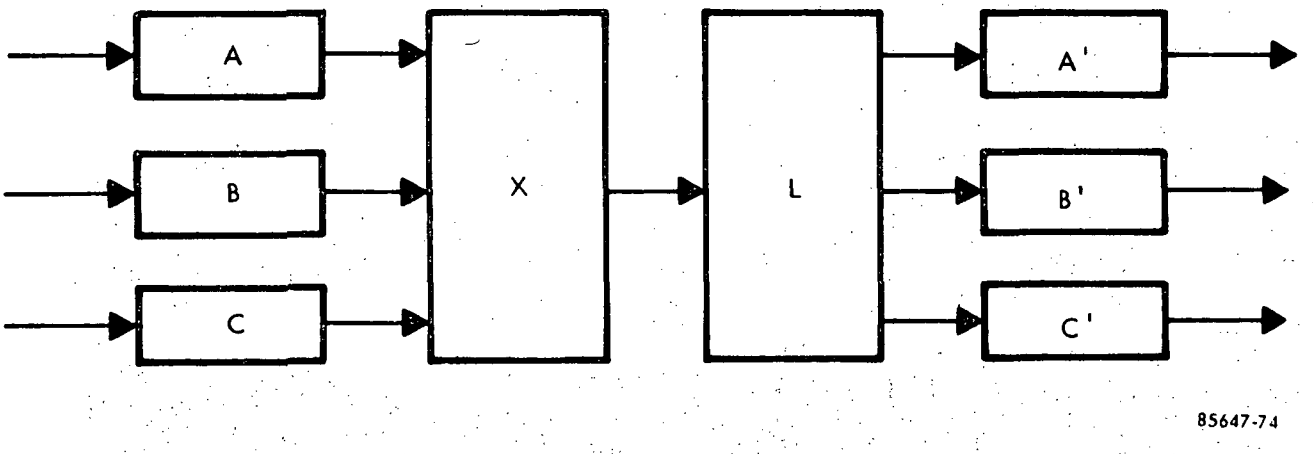


Figure 3.2.7.4. Example of Logical Deduction

The one general statement which can easily be made concerning logical deduction is that it will, to the utmost, be specialized to the equipment it is designed to verify. The advantages gained in reducing the number of observation points and the attendant reductions in observing equipment will be traded off against loss in flexibility. If logical deduction is chosen for extensive use in a complex system, any change in system configuration could lead to a major revision of the verification scheme.

The time required for a complete verification process under logical deduction will be less, in some cases far less, than that required by iterative deduction.

Logical deduction has the potential of being the least costly of the three deduction schemes from a hardware standpoint. Unfortunately, its applications are limited and each savings of a verification point must be replaced by logic circuitry. One important application of logical deduction is found in verification aboard a spacecraft. Here, weight, power and space are all at a premium and logic operations can usually be performed on a computer (if the requirements are identified early). Logical deduction under these circumstances could prove quite promising.

Using the guidelines set forth in this section, a deduction scheme can be selected for each group in the system. If iterative or logical deduction (or a combination thereof) is used, logic and control must then be developed. This is the subject of the following section.

3.2.7.5 Logic and Control for Group Schemes

BLOCK D.11

Iterative and logical deduction schemes will require special control and logic to be implemented. It is not the intent of this section to describe how logic design is implemented but rather to point out the need for this requirement.

Iterative deduction will operate from status indications provided by the verification equipment at the output of the group. When an unacceptable indication is received, sequential commands must be sent to the various principal system reconfiguration switches. This operation will involve drivers and timing to be implemented. The switching sequence is seldom arbitrary since algorithms must be devised to minimize the average switching time. The algorithms must be implemented into hardware to provide control of the operation. An accounting system must also be included so that configuration changes can be tracked and status of the proper element indicated.

Logical deduction will involve the processing of simultaneous verification inputs such that conclusions may be drawn based on various combinations of status indications. This involves the implementation of logic statements and algorithms for processing. If the logic is extensive, it is best handled on a real time computer.

Once deduction logic and control have been designed, the method of internal status determination is complete. However, deduction schemes depend on group verification results and the design process must rejoin the set problem at this point to determine an approach to external verification.

3.2.8 Approach to Off-Line Elements

BLOCK D.12

It is appropriate to begin by offering a distinction between on-line and off-line elements. Consider a set in which the element outputs are isolated by a switch. Further, only one of the elements in the set is in the information stream of the principal system at any time, i.e., throughputting the tenant signal to successive functions. The remaining elements have their outputs dead ended. The one element which throughputs to successive functions is called the on-line element and the remaining elements, off-line elements.

Thusfar in the handbook, sets and verification techniques have been described in what may seem to be a one-to-one correspondence. This is not necessarily true. Under the situation in the preceding paragraph, it is entirely possible that one verification scheme could be used for the on-line element and a different technique for the off-line element. This was evidenced in the description of iterative deduction in Section 3.2.7.4 and points up one scheme for treating off-line elements.

A second, and more obvious method of treating off-line elements is to simply let each be an entity and verify it on its own basis. This can be accomplished in two ways: (a) perform verification while the element is operating on the input tenant signal, or (b) inject a simulative signal into the element. If the element contains entities which must be updated in time (stored data, time variable logic, etc.), approach (a) is the only feasible solution. If the redundancy is nonsymmetrical or if the capabilities of the elements differ, (b) is usually the logical choice.

There are other schemes applicable to this problem – especially if there are two off-line elements in a set. Without pursuing these; however, it can be seen that a policy regarding the treatment of off-line elements is essential at the outset of a design. Perhaps the most important point to be made in this section is that *there is no reason why verification of on-line and off-line equipment should employ the same techniques.*

3.2.9 Summary of Design Data

BLOCK D.13

The intent of Section 3.2, et seq., has been to define the verification problem and establish design requirements for all items to be verified. At this stage in the design process, the items listed below should be identified and quantified if possible. It is not expected that all the items will represent concrete design constraints. Some must simply be positions to be held until more detailed information becomes available. The important point is that the best possible design posture be represented so that subsequent design effort will have a baseline from which to move forward.

The following items should be identified at this stage of the design.

a. All groups and sets

b. All simulative signal injection points (with signal specifications where possible)

c. All IBV's and their associated verification points (IBV's may be elements, sets or groups.)

d. For each IBV:

- Whether real or deferred time verification is required
- Verification rate for deferred time verification
- A general statement of overall confidence in verification results for each level of confidence
- Mission time
- Failure notification time for each level of confidence
- Probability of Type I and Type II error for each level of confidence
- Signal characteristics at each verification point

With these data identified, it will be possible to begin design of the individual IBV status development approaches. This is the subject of Section 4.0. Before departing Section 3.0, however, two additional areas must be explored. First, it will be necessary to identify the areas of interface between the principal system and the verification system.

This is discussed in Section 3.2.10. Second, if the design requirements identified above are to be achieved in a reasonable fashion by status development, the impact of design limitations imposed by status development should be considered. This is the subject of Section 3.3.

3.2.10 Identifying Interfaces

BLOCK D.14

The verification design must obviously interface with the principal system. If verification equipment is to be effectively designed, these interfaces must be considered as part of that design. It is impossible to *specify* interface requirements at this stage in the design process and the intent of this section is to identify the interfaces so they may be followed up as design progresses.

An interface will exist at each signal injection point. The principal system characteristics at these points must be known and allowance made for switching.

Interfaces will exist at each verification point. Allowances must be made in principal system design for signal extraction at these points.

Much of the verification equipment must be physically adjacent to its associated IBV. Mechanical and electrical interfaces will exist and allowance made in the principal system. Verification equipment should be compatible with packaging approaches in the principal system. Additional wire routing and connectors must be incorporated into the principal system.

Verification equipment will require power and cooling. To the maximum extent possible, the verification design should be compatible with these factors existing in the principal system.

Verification equipment may well require time sharing on a central processor. If this can be incorporated into an existing principal system processor, estimates must be made regarding processing time and data rates. The verification equipment must have a compatible format.

3.3 Impact of Status Development Requirements

BLOCK J.0

Most of the impacts resulting from status development already appear throughout Section 3.2. This section will then summarize these factors for reference.

Status development extracts selected properties of signals present at the output of an IBV, performs any necessary processing of these properties and makes status decisions based on values of the properties. The most significant impact area will be that of status confidence resulting from the amount of information available in the IBV signals and the error probability of the decisions.

If sufficient confidence cannot be achieved using tenant signals, attempts at using symbiotic signals must be made. If this is not possible, deferred time verification using simulative signals will likely be required. An alternative to using injected signals is to redefine the IBV such that the verification point is altered to a position which is more amenable to verification. This may be the least costly approach (assuming it is possible) but will impact partitioning.

Once signals and signal properties are established at an IBV output, status development errors can impact characterization results. If error probabilities are too great, real time verification may have to give way to deferred time verification or failure notification times may have to be increased. If deferred time verification is being used, excessive error probabilities may influence a decrease in verification rate.

These factors must be kept in mind while identifying and defining items to be verified. Imposing too stringent requirements on error probability, notification time, etc., can only result in design recycles. An appreciation of the status development problem is the best way to forestall this problem.

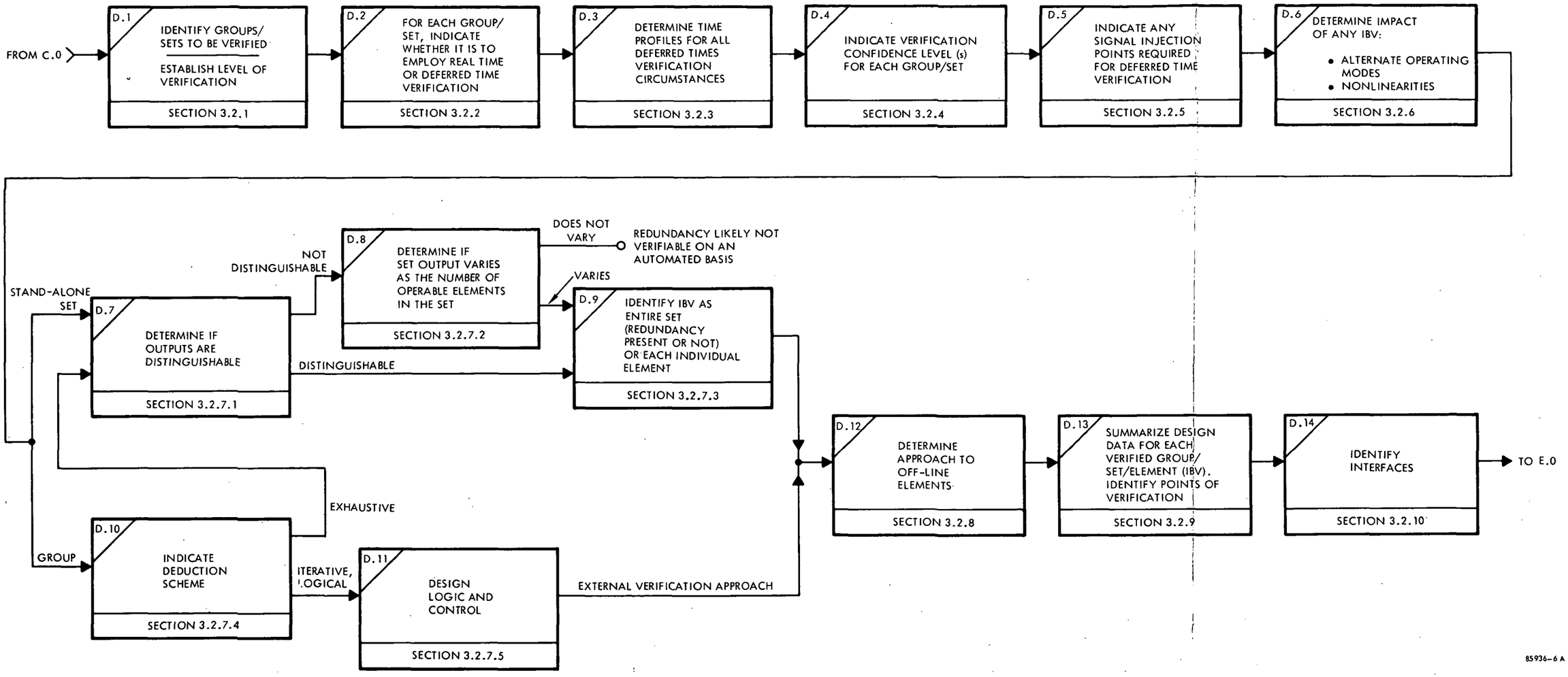
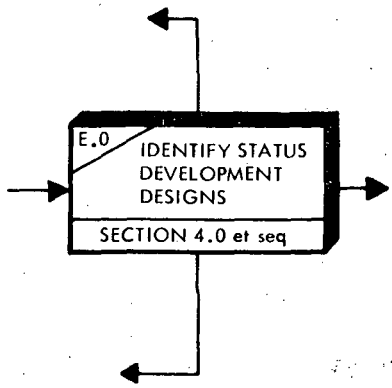


Figure 3.2. Characterization Design Process

SECTION 4.0

DESIGN OF STATUS DEVELOPMENT



4.0 DESIGN OF STATUS DEVELOPMENT

Section 3.0 has provided the methodology for identifying those equipments, or collections of equipment, in the principal system which will require points of verification. Each verification point must provide an automated status indication for the Item Being Verified (IBV) and is located at the output of the (IBV). This automated determination of status is known as *status development* and its achievement is the subject of this section.

A status development operation will be used at each point of verification i.e., for each IBV, in the principal system. The function of status development is to indicate the operational integrity of an IBV associated with a point of verification. The indication of operational integrity will take on the form of discrete indications such as "go, no-go" or "green, amber, red", where these gross operational descriptors are called status.

If status is to be indicated on an automated basis, information must be derived from the IBV which serves as an indicator of its performance. However, performance indications by themselves will not be satisfactory. If status is to be determined, these indications must be judged as satisfactory or unsatisfactory and this can only be accomplished with the use of a reference. *Each implementation of status development must then contain some form of reference for purposes of comparison.*

Once performance indications are compared to a reference, an indication of "goodness" or operational integrity is achieved. This indication is called a status variable and must be converted to status. The conversion involves essentially a second form of reference in that values of the status variable must be compared to fixed ranges corresponding to the levels of status. This process is depicted in Figure 4.0–1. Shown, is a status variable and the associated ranges corresponding to the levels of status. The process of converting the continuous status variable into discrete levels is called the *mapping operation.*

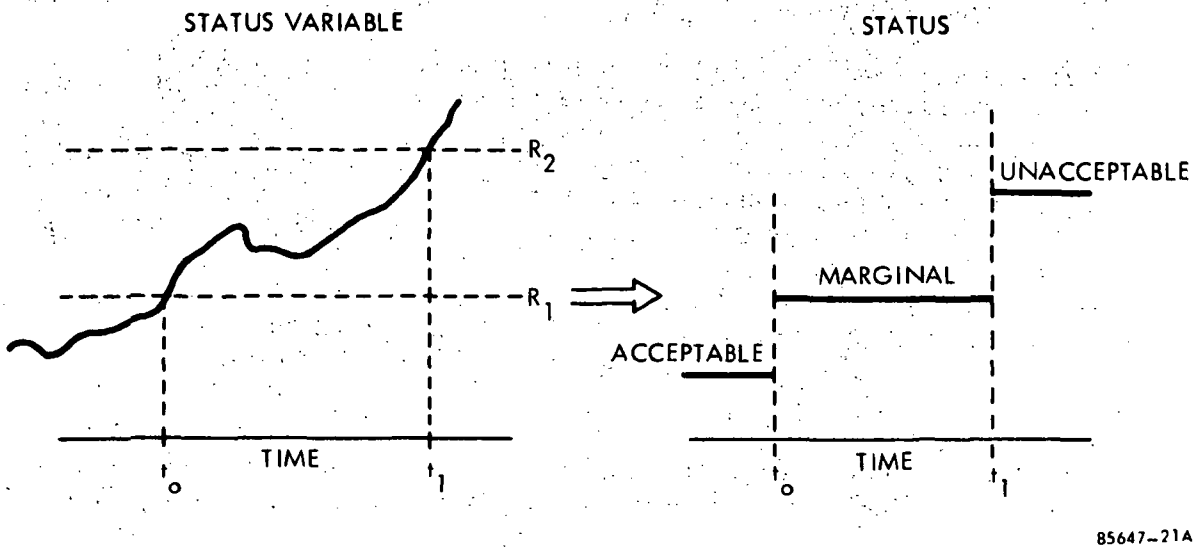


Figure 4.0–1. Relationship of the Status Variable to Status

With the general description above, it will now be advantageous to increase the degree of complexity and view the operation from an implementation standpoint. The first step is to identify what properties of the IBV will best provide an indication of its operation. Depending on the complexity of the IBV and the degree of confidence desired, the number of different properties may range from one to a dozen. These properties are called *performance criteria* and are typically extracted from the IBV output. Two such criteria might be average signal amplitude (in volts) and signal instantaneous frequency.

Once the performance criteria are selected, they must be extracted or otherwise developed from the IBV output signal in the form of voltage analogs. In the examples above, signal voltage must be averaged and a descriminator might well be used to determine instantaneous frequency.

Once voltage analogs are achieved, it will be necessary to compare these values against some reference to determine how far they depart from the recognized norm. The resulting quantity is called the performance measure.

Implementation requirements will usually dictate the addition of an operation not identified in the general discussion. This is a *smoothing operation.* If the performance measure were to be used directly as a basis for status determination, random fluctations (which are certain to be present) could cause status errors. The smoothing operation reduces these fluctations, thus decreasing the likelihood of error. The smoothed performance measure is the status variable.

The status variable must be converted to discrete values representing statements of status in the mapping operation. This operation must decide, based on the value of the status variable, which level of status to indicate. Decisions of this nature will involve thresholds to bound ranges of the status variable, e.g., see R_1 and R_2 in Figure 4.0–1. Mapping is then a threshold operation which implements decision rules elected by the designer.

Figure 4.0–2 illustrates an implementation of status development. Shown, are the operations to be performed using circuits that could realize each operation. *It should be recognized that, typically, a similar network will be necessary for each performance property of interest.* If a principal system consists of twenty IBV's and each IBV requires five distinct properties to be examined, there will be a total of 100 such networks.

The preceding discussion assumes that analog operations will be used throughout. This has been adhered to in these introductory comments since the operation is more easily viewed from the analog standpoint. The same operations will apply to digital signals, but, of course, the implementation will be different. The reader desiring additional detail should consult Reference [2].

The process for designing status development is shown in Figure 4.0–3 which is included as a foldout on Page 111 at the end of Section 4.0 *et seq.* The process essentially follows the descriptive development presented above, with one exception. There is a feedback block (Block E.9) from "determination of decision rules" back to "selection of performance criteria." Definitive decision rules cannot always be identified for performance criteria. If such a criterion is selected, it will either have to be altered or deleted.

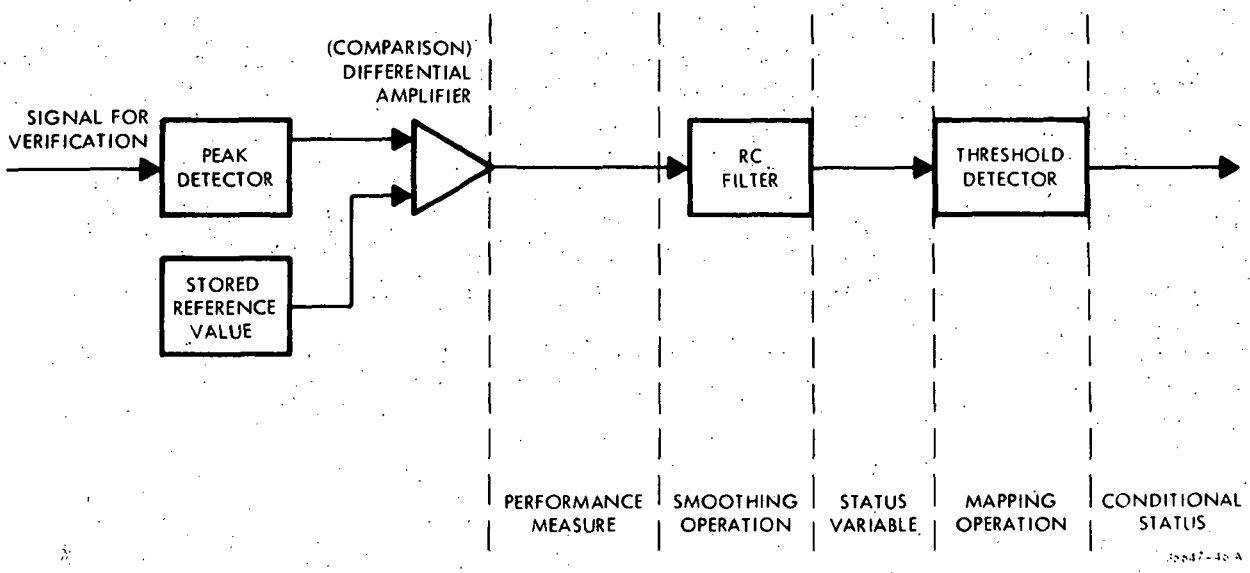


Figure 4.0–2. Example of Status Development

As a final introductory comment, this section touches on a rather wide variety of subjects and technological disciplines. As the material is somewhat condensed, the reader may find it helpful to refer to the sample implementation contained in Section 4.7 from time to time.

4.1 PROPERTIES INDICATIVE OF OPERATION

BLOCK E.1

Identifying properties indicative of IBV operation is probably the most difficult and least definable task in status development. In effect, it amounts to identifying those features which will satisfy the designer that the IBV is operating properly. These features will obviously be quite different for different types of hardware and will be altered even further by different applications of the same piece of hardware. For example, a performance criterion for a telemetry link might be error rate whereas a guidance system might use average error over a two hour period.

The number of performance criteria required for an IBV will usually depend on its criticality to the mission and its complexity. One would not be as critical of an angle encoder for an automobile assembly line as for a tracking antenna. The latter may well require additional criteria. While firm statements cannot be made regarding selection of performance criteria, some guidelines will be beneficial.

a. The criterion must be measurable, i.e., quantifiable and lend itself to instrumentation. It will be of little value to identify a performance criterion which cannot be reasonably represented by numbers or measured on an automated basis.

b. The criteria should represent the most significant properties of performance – the most critical, the ones of greatest interest – for the function being performed. A designer is often motivated to increase the number of criteria to increase his confidence that the operational integrity is being

accurately portrayed. The most significant factors motivating a *decrease* of the number of criteria are cost, reliability, complexity and diminishing returns. Each new criterion requires additional decisions to be made, inviting the inevitable increase in decision error. One can rapidly find himself being more and more precise with less and less accurate information.

c. In the interest of efficiency and accuracy, the criteria should be as mutually independent as possible. It would be desirable not to have to make two decisions on the same piece of information. This oftentimes conflicts with measurability and ease of implementation mentioned in (a) above.

d. This final guideline is not as much concerned with performance criteria as with means of implementing them. Measurement of a physical property (extraction of the performance criterion) at the IBV should be free of influence from other variables in the IBV. If longitudinal acceleration of a booster is being determined, the accelerometer should be mounted such that it is as free as possible from influence of roll or transverse acceleration.

It should be borne in mind that criteria may change as IBV modes of operation change. Different sets of criteria may have to be identified for each mode.

4.2 REAL TIME AND DEFERRED TIME

The criteria identified in Section 4.1 will rarely be amenable to real time verification. It is possible that the tenant signal(s) will not contain enough information or that a symbiotic signal cannot be injected. If the criterion is mandatory, it must then be measured deferred time.

Decision rules can also influence the selection of criteria or affect their time relationship. An identified criteria could well cause problems when decision thresholds are to be identified.

4.2.1 Establishing the Relationship between Time and Performance Criteria

BLOCK E.2

With respect to time, performance criteria can be thought of as falling into three categories. Those which can be determined:

a. real time with high confidence,

b. real time with reduced confidence followed by deferred time high confidence, or

c. deferred time with high confidence only.

Using the signal characteristics and IBV functional descriptions supplied from Section 3.0, it is possible to relate the performance criteria to available signal properties (or mechanical effects) and thus determine the time relationships. The ideal situation is when all criteria

can be determined from tenant signals in real time with high confidence. If it is required that an IBV be verified real time and the criteria cannot be determined real time, the difference must be traded off.

Category (b) above offers a compromise. Status development for many IBV's will fall into this category. Status of an IBV can be determined with an acceptable confidence during real time to detect any radical problems. High confidence, deferred time verification will then be used to follow up with a more thorough check. Use of this category will immediately double the design effort, however, since both deferred and real time problems must be solved.

4.2.2 Impact of Decision Rule Factors

BLOCK E.9

If a mapping operation is to be practically implemented, the *threshold values must be constant and not referenced to any other principal system property.* For example, implementing a threshold which considers the status of an IBV acceptable, so long as the status variable is four volts above the IBV input value, is neither efficient nor readily implementable. As stated, the indication is perfectly acceptable as a performance criterion, but the status variable should represent the difference between the input and the property of interest. The threshold can then be set at a constant four volts.

It must be possible to identify a decision rule which has physical significance. Instantaneous gas mileage on an automobile can be measured, but what constitutes acceptable operation? If the performance criterion were changed to average gas mileage, one could readily identify the value above which acceptable operation is considered.

When selecting performance criteria or identifying their time relationships, the designer must keep in mind the requirements of decision rules. It will accomplish little to identify a criterion for which no decision can be made regarding acceptable operation.

4.3 REQUIREMENTS FOR REAL TIME INJECTED SIGNALS

BLOCK E.3

By introducing a symbiotic signal (see Section 3.2.5) into the IBV, it may be possible to determine an otherwise deferred time criteria in real time. The symbiotic signal would then complement information present in the tenant signal. Not all IBV's or tenant signals are amenable to the addition of symbiotic signals and their introduction can force changes in the principal system such as increased bandwidth, added coding bits or alteration of data frame formats. In any event, if a signal can be devised, its injection point must be identified and necessary design changes to the principal system identified and coordinated.

The requirement for a symbiotic signal frequently arises when the property of interest is not deterministic (see following paragraph). The symbiotic signal, since it is under the control of the designer, will be deterministic and allow the identification of a decision rule.

The idea of deterministic properties will be used frequently throughout the remainder of this section and it will be well to elaborate. A performance criterion for which a decision rule can be stated at any point in time is termed a deterministic criterion. For brevity, the property or signal (in the IBV) which is the measure of a deterministic performance criterion, will be termed a deterministic property or signal. This convenience can lead to misunderstanding if the above distinction is not kept in mind. *When a property or signal is deterministic, one can state what its value (or more precisely its range of values) should be at any point in time.* This is not meant to imply that the actual, measured property will be predictable; for if it were, equipment failure could be predicted. For example, the voltage output of a 10 Vdc power supply is called deterministic since one may state that this value *should* be, say 10 $\pm$0.2 Vdc for all time. The output of an AM radio is stochastic (not deterministic) in that very little can be said about its value (in an absolute sense) over time.

Consideration of deterministic properties adds one more restriction to mapping threshold previously discussed in Section 4.2.2. That is, *a mapping threshold must not only be restricted to a constant value but to operation on deterministic performance criteria as well.*

4.4 DEVELOPMENT OF PERFORMANCE MEASURE

BLOCK E.4

With the identification of performance criteria, the design must now turn to implementing these criteria in the form of performance measures. This involves three distinct operations: extracting the measure of interest from IBV outputs; establishing a reference on which to base operational integrity and performing any required special operations such as averaging.

4.4.1 Extracting Measures of Properties

Extracting measures of properties is typically realized with transducers in mechanical systems. In electronic systems, these measures are typically realized by extracting relevant properties of IBV output signals. These output signals may be tenant, symbiotic, idle or simulative signals depending on the circumstances. The measure thus developed is also a signal and is usually a voltage analog of the property of interest. An example is a discriminator which produces a voltage proportional to frequency deviation. If limited spectrum information is desired, a notch filter which can provide the signal energy within its bandpass would be another example of extraction.

The point to be made in this section is that some preselection or transducer will be required at the IBV output to obtain the initial measure desired. The desired measure can always be determined from the performance criterion. For example, consider the criterion: average frequency deviation. The measured property would be frequency (quite likely instantaneous frequency). Since an average is sought, the measured property will have to undergo averaging at a later point before the final performance measure is realized. This operation is the subject of Section 4.4.3.

4.4.2 Establishing a Reference

A measure must have a bench mark or frame of reference. Indicating that a signal is four volts without indicating a reference for the measurement is only half the story.

There are four basic ways of establishing a reference for status development.

- IBV output signal properties and their absolute value.
- IBV output signal properties as they relate to IBV input.
- IBV output signal properties as they relate to like properties of outputs from other elements in a set.
- Signals (both input and output); as they relate to IBV operational description.

The performance criterion should contain the key as to which method to use. Consider an oscillator which is to provide a reference frequency of 4 KC. This value is stated in absolute terms, i.e., it implies no comparison to any other property in the system. A voltage analog of the frequency (the signal property of interest) output can simply be compared to a constant voltage to determine performance. Such a device would employ the method of output signal properties and their absolute value. (Note that the output signal frequency is a deterministic property.)

The above example points out another important consideration. Since the property measure will require comparison to a constant value, there is actually no need in forming a difference to realize the performance measure. The analog of instantaneous oscillator frequency can be compared directly with a threshold in the mapping operation. *Status can thus be indicated directly from the relationship of the performance measure to the threshold. This is characteristic of reference methods using absolute values.*

Consider next, an analog amplifier whose function is to provide a faithful reproduction of the input amplified by A. It is quite difficult to translate "faithful reproduction of the input" into absolute terms. Such a device would likely use the method of output signal properties as they compare to IBV input. In effect, it would not be possible to express the amplifier output in deterministic terms without referencing the input.

It is entirely possible that an absolute statement cannot be made about an IBV output nor can a simple relationship be established to the input. If this is the case, the designer might be satisfied with knowing the output of one element in a redundant set agreed with the other element. This would employ the method of comparing like properties of the outputs of other elements in a set. An obvious disadvantage to this method is that if a disagreement occurs between two IBV's, it will not be possible to determine *which* is correct.

The final reference method is quite sophisticated and, essentially, attempts to compare the specified IBV transfer function with one developed using input and output signals. The method has found application in automatic long line equalization techniques.

The reference methods are shown in Figure 4.4.2 along with techniques for implementing the methods. An extensive discussion on each technique is contained in Appendix A. With the aid of this appendix, the designer should be able to systematically select the proper technique based on the performance criterion. Note that the two value check techniques and spectral analysis must be used with deterministic signal properties. For the items identified as being applicable to stochastic properties, it is necessary to point out that resulting performance measures must be deterministic, even though the techniques are suited for stochastic signal properties. For example, signal form analysis determines statistics (e.g., average value) from stochastic property measures (voltage analogs of a signal property) but the generated average value must be deterministic if the mapping operation is to be implemented.

In addition to the reference methods identified, there is an additional verification technique which does not truly qualify as a reference. This technique is called acknowledgment. The technique finds wide application in command and control systems and operates on the principle of feedback. For example, if a valve is commanded to close, a signal is fed back to the device initiating the command indicating the closure has occurred. If the closure did not occur, all the hardware from the device initiating the command to (and including) the valve (all of which would be the IBV) is identified as functioning improperly.

While far from an automated technique, user complaint should be identified for completeness. When all else fails, the user will inevitably discover that a system has failed.

The general applicability of the various techniques is summarized below.

- Monitoring techniques (nonsequential value checks, sequential value check, and spectral analysis) cannot be applied to stochastic signal properties. This is because these techniques demand comparison against stored information which represents what is expected of the output.
- Compare-two and cross-power spectrum techniques cannot indicate the status of each element in a set.
- Coding, voting, cross-power spectrum and compare-two techniques cannot be used when the outputs/effects of each element are not distinguishable.
- Neither acknowledge nor user complaint techniques can be used if outputs/effects of each element are not distinguishable or if continuous verification is required.

4.4.3 Identifying Special Operations

This section will identify several important operations and relate the significance of each to a performance criterion. A description of the special operation will

REFERENCE METHOD	TECHNIQUE	(SEE NOTES)	TYPICAL OPERATION
OUTPUT SIGNAL PROPERTIES AND THEIR ABSOLUTE VALUES	SEQUENTIAL VALUE CHECK	(D)	MEASURE OUTPUT SIGNAL PROPERTIES AND COMPARE TO ORDER-REQUISITE REFERENCES.
	NONSEQUENTIAL VALUE CHECK	(D)	MEASURE OUTPUT SIGNAL PROPERTIES AND COMPARE TO POINT REFERENCES.
	CODING	(S)	DECODE AND COMPARE WITH REFERENCE VALUE/MATRIX.
	SIGNAL FORM ANALYSIS	(S)	DEVELOP OUTPUT SIGNAL STATISTICS ON SPECIAL OPERATION AND COMPARE WITH A POINT REFERENCE.
	SPECTRAL ANALYSIS	(D)	EXTRACT FREQUENCY INFORMATION FROM OUTPUT SIGNAL IN TERMS OF ENERGY IN BANDS AND COMPARE WITH REFERENCES.
OUTPUT SIGNAL PROPERTIES AS THEY RELATE TO IBV INPUT.	INVERSE TRANSFORM	(S)	COMPARE PROPERTIES OF INVERSE TRANSFORMED IBV OUTPUT SIGNAL TO INPUT SIGNAL PROPERTIES.
OUTPUT SIGNAL PROPERTIES AS THEY RELATE TO LIKE PROPERTIES OF OUTPUTS FROM OTHER ELEMENTS IN A SET	COMPARE-TWO	(S)	COMPARE LIKE PROPERTIES OF TWO ELEMENTS IN A SET.
	VOTING	(S)	COMPARE LIKE PROPERTIES OF THREE OR MORE ELEMENTS IN A SET. SELECT VALUE OF MAJORITY AS REFERENCE.
	CROSSPOWER SPECTRAL ANALYSIS (1)	(S)	DEVELOP COHERENCE FUNCTION, $\overline{\alpha}^2$, AND COMPARE WITH A POINT REFERENCE. (2)
SIGNALS (BOTH INPUT AND OUTPUT) AS THEY RELATE TO IBV OPERATIONAL DESCRIPTION.	CORRELATION, PSEUDO-RANDOM NOISE (1)	(S)	CROSSCORRELATE IBV OUTPUT AND RANDOM INPUT. COMPARE RESULT WITH REFERENCE IMPULSE RESPONSE FUNCTION, h(t).
	FREQUENCY TRANSFER FUNCTION (1)	(S)	DEVELOP CROSS POWER SPECTRUM OF IBV INPUT AND OUTPUT AND AUTOSPECTRUM OF INPUT. CALCULATE THE FREQUENCY TRANSFER FUNCTION, $H(\omega)$ AND COMPARE WITH REFERENCE FREQUENCY TRANSFER FUNCTION. (3)

(1) EXCLUSIVELY COMPUTER IMPLEMENTED.

(2) $\overline{\alpha}^2 = \dfrac{\left|\overline{G_{xy}}\right|^2}{\overline{G_{xx}}\,\overline{G_{yy}}}$

(3) $H(\omega) = \dfrac{G_{IO}(\omega)}{\overline{G_{II}}(\omega)}$

THE LETTERS APPEARING IN PARENTHESIS INDICATE THE STATISTICAL CHARACTER OF THE SIGNAL PROPERTY TO WHICH THE TECHNIQUE IS SUITED.

D - DETERMINISTIC ONLY

S - STOCHASTIC, MAY BE USED FOR DETERMINISTIC BUT POSSIBLY NOT EFFICIENTLY

85936 12 A

Figure 4.4.2. Summary of Methods for Achieving Reference

always appear in the criterion statement; e.g., *mean* difference of two properties. It should be noted that a reference can be achieved either before or after the special operation has been performed. In the preceding example, the special operation was performed *after* achieving a reference since the mean is determined for a difference quantity. The results presented in the following sections may be applied to either order of occurrence.

Some performance criteria will depend on the nature of the signals being processed by the IBV and, from the application standpoint, implementation of special operations will certainly depend on the nature of the signal. For these reasons the material has been divided into the three major groups of analog signals, digital signals and sampled data signals. It is recognized that these signals do not represent an exhaustive treatment of all possible signals; however, it does account for those most widely used. Treatment of signals not described here will many times involve only an extension of these results to that particular case.

4.4.3.1 Analog Signals

The operations discussed in this section have been identified with analog signals because they find their greatest application here. Some of these operations could also be used with sampled data or digital signals under special applications. For example, in conjunction with a sample and hold, virtually all could be applied to sampled data.

With the addition of a sample-and-hold and an A/D converter, all the identified operations can be performed on a digital computer. This would, however, amount to sampled data input and the comments of Section 4.4.3.3 would apply.

The reader's attention is called to several sample functional implementations in Section 4.7 which exemplify the material in this section.

Performance Criterion of Peak Value

This performance criterion imposes the special operation of peak detection. Concern may be with maximum or minimum peaks but typically the former. Peak detection may be used on sampled data as well as analog signals. Peak detection can be used to detect the peak value of a periodic signal or a nonperiodic signal. In the latter application, a period of interest must be established, i.e., concern must be with the peak value reached during some period. Under this application, the output of the detector will be discrete (of the sampled data type). Peak detection will furnish no information as to how the peak value was achieved. Circuitry for the detector will typically involve a diode and a capacitor. Decay time will be important and without special provisions, response time will be proportional to decay time, i.e., the output can be effectively delayed. Critical applications will require a current controlled source for the capacitor.

Performance Criterion of Average Value

This performance criterion imposes the special operation of averaging which can be implemented by passive or active low pass filters or by an integrator.* Filters have the advantage of continuous output but, without multiple pole designs, the possible disadvantage of a larger output variance due to wider bandpass. Integrators are active devices and will require a power source. Practical applications of integrators will require a fixed integrating time period; consequently, they require timing and the output is of the sampled data type. The material in Section 4.5 is directly applicable to these averaging devices and will not be repeated here. Suffice it to say that in all cases, the averages are being performed over time. The final design is a trade-off of variance in the estimate of the average and response of the estimate to changes in the true mean on the input.

Performance Criterion of Integral Squared Difference (Or Error)

This performance criterion imposes the special operation of squaring the input and integrating this quantity over a fixed time period. The criterion is only suitable if a reference (difference or error) is formed beforehand. The integrator will require timing and the output will be of the sampled data type. This criterion finds application where not only magnitude but duration of a difference is important and the sense of the difference is of no consequence. The criterion, due to squaring, tends to weight large differences more heavily than small differences. As an application example, consider a closed loop control system. The error signal affords a good indication of its performance – not only lag and dynamic response to step inputs but also null response due to noise. It would not be reasonable to penalize the system by using the magnitude of error due to dynamics and since it must respond to inputs of any sense, sign of the error should not bias the result. Integral squared error is a reasonable performance criterion for this system.

Performance Criterion of Squared Error

This performance criterion imposes the special operation of squaring the input value. Like the integral squared error, it must have a difference formed beforehand to be meaningful. The similarity continues in that large errors are weighted more heavily than small ones and the sense of the error or difference is lost. This criterion would be used where magnitude of error is important and where this magnitude should be weighted. Implementation would likely be with a quarter-squared multiplier. This is its biggest disadvantage since it involves a sizable amount of active circuitry. Another disadvantage is that the squaring process will introduce more variability into the resulting performance measure. This variability must be reduced by the smoothing operation.

Performance Criterion of Absolute Value Error

This criterion is similar to squared error except the magnitude of error is not weighted by squaring. Virtually any implementation of this operation will involve active circuits.

---

*As used here and throughout the handbook, the term "integrator" means an integrating D.C. amplifier.

Two common implementations are (a), a bridge rectifier with differential sense and isolation circuitry, and (b) an arrangement of diodes in conjunction with two D.C. amplifiers. Both applications will provide continuous output and both are subject to small signal error from the forward voltage drop on the diodes.

4.4.3.2 Digital Signals

It is appropriate to begin by defining what is meant by a digital signal. Simply stated, a digital signal is a signal which is discrete in both time and amplitude. Interest will be restricted to two amplitude levels or the binary case.

For digital signals which represent digitized analog signals, any *criterion* in Section 4.4.3.1 can be considered applicable*. The indicated operations would most likely be performed on a computer and the material of Section 4.4.3.3, Sampled Data Signals, and Section 4.5.3 would apply. The averaging techniques described in Section 4.5.3 are considerably more efficient than performing the digital counterpart of integration. In addition to sampling errors, the designer must also account for digitizing and processing errors.

While performing the operations above is certainly meaningful, a much more efficient and reliable means of verifying digital systems is available. This is due to the unique ability of a digital system to indicate an error with no *a priori* knowledge required as to the *value* of the digital word. This is achieved with the use of redundancy bits or by coding. Considering this feature, the most valuable performance criteria for a digital IBV would deal with bit error rate.** There are two such criteria in common use. The first is concerned with the number of consecutive errors committed (assuming all errors are detected) and is mostly applicable to machines with high bit error rates or under circumstances where degradation of error rate is considered negligible compared to a "hard" failure. The procedure begins by initiating a count when an error occurs and continues counting consecutive errors until no error is observed. The sequence will begin again whenever the next error occurs. If the count ever exceeds the established decision boundary, the IBV operation is declared unsatisfactory. As an example, consider an IBV which has a word error probability of 0.01 (when performing according to specification). Establish a decision rule that if five or more consecutive errors are made, the status will be declared unacceptable. If a failure occurs (assuming all errors are detected), the failure will be detected in five word times. This is a very rapid response and the greatest advantage of this criterion. The next question is concerned with the probability that a good IBV, i.e., one operating with the above probability of error, will be declared unacceptable by this criterion. This is the probability of committing

---

*Also, there is no reason why, say, an average may not be meaningful when calculated on a series of binary digits. This would employ the methods of Section 4.4.3.1.

**Where error correction and detection are used, error rate, for the purpose of this handbook, will include errors detected but not corrected. If a machine commits four errors on a word and three are corrected, only the uncorrected error would contribute toward determining bit error rate.

five errors in succession for a single trial, which is $(0.01)^5$ or 1×10^{-10}. This is the probability of a false alarm, assuming an error-free decision device.* While the number is quite small, it should be noted that the average word error rate could drift quite high and the likelihood of this criterion detecting this condition would still be small. For example, if word error probability reached 0.1, the probability of detecting this condition is 1×10^{-5}. This quantity is still quite small and is considered to be a disadvantage of this criterion. It is then worthwhile to consider the next criterion which measures bit error rate.

The second criterion of interest counts errors over a given period and decisions are made on these counts for each period. As in the criterion above, interest again is centered on the confidence in the count and the time required to make a decision. Confidence in the above criterion was a simple matter; however, under this criterion, the probability distribution must be known before confidence statements can be made. If errors are assumed equally likely for all bits, a good approximation can be made of this distribution.

Using the criterion of errors per period of time, the designer is given:

- the IBV design bit error rate,
- the highest bit error rate which is still considered to be acceptable operation,
- the IBV bit rate, and
- a confidence number representing the probability that the verification equipment will not inadvertently identify as unacceptable, an IBV which is actually operating at the design bit error rate.

A decision rule must then be formulated which meets these requirements. Determination of this decision rule involves solving for two unknowns: the duration of the observation period and the maximum number of allowable errors which may occur during the observation period. This latter unknown forms the decision threshold.

The exact distribution of the number of errors observed in a time interval T is binominal and can be described with the following random variables and parameters.

n = a random variable which is the observed number of errors in any particular observation interval.

M = the number of bits examined in any particular observation interval.

N = the expected or average number of errors in an observation interval based on previous knowledge.

---

* This brief analysis is intended to demonstrate the operation and possible shortcoming of the criterion. The analysis is not complete. A detailed analysis is to be found in Section 4.8.1.

T = the time length in seconds of an observation interval.

br = the designed bit rate in bits per second.

dber = the designed bit error rate which is the probability of a bit error.

mber = the maximum allowable bit error rate which sets the upper limit on the probability of a bit error.

ober = a random variable which is the observed bit error rate.

CL = a confidence level, $0 \leq CL \leq 1$.

n_o = maximum number of allowable errors (the decision threshold).

Thus it can be seen that the following relations hold.

$$N = (dber) \cdot (br) \cdot (T)$$

$$M = (br) \cdot (T)$$

$$n = (ober) \cdot (br) \cdot (T)$$

The probability that the number of observed errors, n, is less than or equal to some number, R, is binomial and can be expressed as

$$P_r\left[n \leq R\right] = \sum_{k=o}^{R} \binom{M}{k} \left(N/M\right)^k (1 - N/M)^{M-k}$$

Performing the indicated operations in this distribution can be a tedious exercise. This process is somewhat simplified if the binomial is approximated by a Poisson distribution. The approximation is valid provided the probability of success is much less than the mean which is in turn much less than the number of observations, viz.,

$$\left(\frac{N}{M}\right) \ll N \ll M$$

In terms of bit error rate, this means

$$(dber) \ll (dber \cdot br \cdot T) \ll (br \cdot T)$$

which will be true provided,

a) $dber \ll 1$, and

b) $1 \ll (br \cdot T)$

Performing the necessary manipulations, an equivalent probability statement is,

$$P_r \left[n \leq n_o \;\middle|\; N \right] = CL \qquad (4.4.3.2-1)$$

where n_o is the maximum number of allowable errors and is related to (mber · br · T) as shown in Figure 4.4.3.2–1. Since an integral number of errors must be observed, n_o is the next lowest integer below the value of (mber · br · T). For example, if (mber · br · T) is 4.7, $n_o = 3$. Note that for all values of (mber · br · T) below 2.0, $n_o = 0$. Using the Poisson approximation, Equation (4.4.3.2–1) can be rewritten as

$$CL = \sum_{k=o}^{n_o} \frac{(N)^k e^{-N}}{k!} \qquad (4.4.3.2-2)$$

The normalized results of Equation (4.4.3.2–2) are shown in Figure 4.4.3.2–2. This figure is known as a Thorndike chart.

An example will help in the use of these results. Consider the following problem. An IBV has a bit rate of 500 K bits/sec, and a designed bit error rate of 10^{-6}. A decision rule is formulated such that if the observed bit error rate exceeds 3×10^{-6} the IBV will be declared unacceptable. Further, the designer wishes to be 99 percent certain that an IBV, which is operating at the designed bit error rate, will not be declared unacceptable. How long should the measurement be taken? The given quantities are:

CL = 0.99

dber = 10^{-6}

br = 5×10^{-5}

mber = 3×10^{-6}

mber/dber = 3

The solution procedure is as follows. Select two values of n_o and (mber · br · T) from Figure 4.4.3.2–1 and plot these on Figure 4.4.3.2–2 using the ratio mber/dber (since dber · br · T and mber · br · T are related by the same ratio). These values should be selected such that a line connecting the two points on Figure 4.4.3.2–2 will cross the desired confidence level. At the point of intersection between CL and the construction line, select the next largest n_o. This is the maximum number of errors that can be allowed under the desired confidence. Now, read down from the selected n_o to obtain the value of (dber · br · T). Since (dber · br) is known, T is immediately available.

Using the given information, select two values of (dber · br · T), say 1.0 and 3.0, these correspond to (mber · br · T) values of 3.0 and 9.0, respectively (since mber/dber = 3). From Figure 4.4.3.2–1, corresponding n_o values of two and eight are obtained. The points (dber · br · T) = 1.0, 3.0 and $n_o = 2, 8$ are plotted on Figure 4.4.3.2–2 and connected by a straight line. This line intersects CL = 0.99 and the next higher value of n_o above this

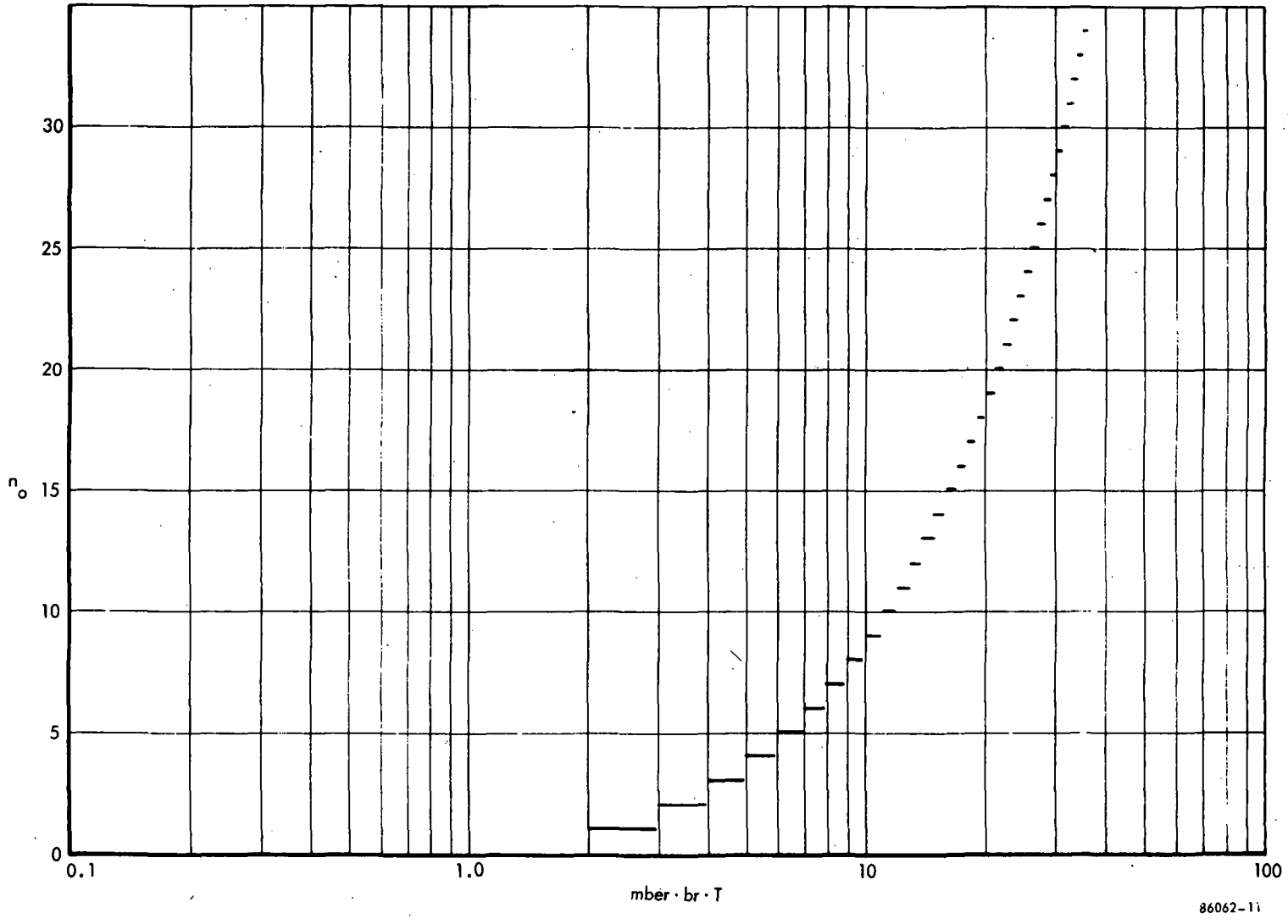


Figure 4.4.3.2—1. Values of n_o for Small Values of mber · br · T

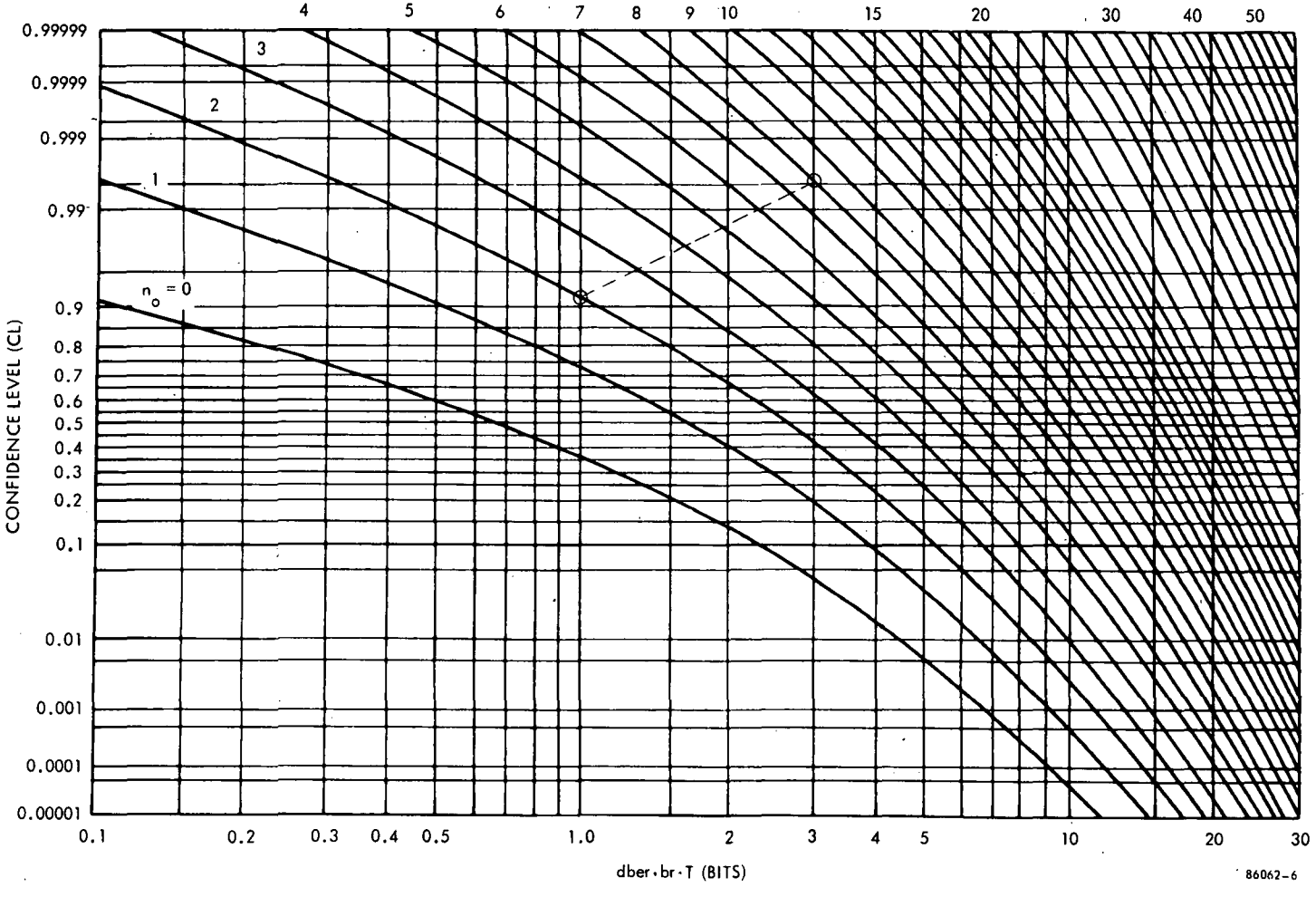


Figure 4.4.3.2–2. Plot of Type I Error Confidence for Determining Sample Time and Number of Errors

intersection (and along CL = 0.99) is $n_0 = 6$. Six errors would then be allowed, seven or more being considered unsatisfactory operation under these conditions. Reading down from the intersection of CL = 0.99 and $n_0 = 6$, find (dber · br · T) ≃ 2.3. Since (dber · br) is 0.5, the observation time, T, is 4.6 seconds.

While the above example demonstrates the use of the curves, there is an important observation to be made regarding the sample or observation time. For systems with very low bit error rate, say 10^{-8}, the observation time can easily become exceptionally long ranging from hours to days unless the ratio of mber/dber is allowed to become quite large. This is obviously an unacceptable solution for detecting failures* and the consecutive error criterion described earlier could well be used in conjunction with this criterion.

It should be noted that the same procedure described above is applicable to word error rates with appropriate substitution of parameters.

Comparing the operations described here for digital signals with those under analog signals, it is easy to draw the conclusion that, for all the reasons digital systems are superior to analog systems, these reasons hold true for verification also.

4.4.3.3 Sampled Data Signals

Sampled data signals are signals which are discrete in time and nondiscrete in amplitude.

There are three basic ways to operate on sampled data:

a. reconstruct the original waveform and perform the operations as indicated in Section 4.4.3.1,

b. operate on the samples themselves, or

c. digitize the samples and perform the operations on a computer.

Method (a) is a straightforward application of Section 4.4.3.1 and will not be repeated here. Emphasis will be placed on method (b) above and each of the *criteria* discussed in Section 4.4.3.1 will be discussed here as well. Unique applications of method (c) will be identified. The output of all operations discussed will also be of the sampled data type. As such, they will all require timing.

Performance Criterion of Peak Value

To operate directly on samples under this criterion will require that a fixed interval be established such that interest is centered on a peak value over this interval, i.e., for an arbitrary signal, it makes little sense to talk about the peak value over all time -- especially if one must wait that long for an output. It is quite likely that a peak

*The reason behind this statement is that the period of observation establishes how frequently decisions are made. By definition, a decision can be made only at the end of a period and one remains ignorant of the status during this time.

in the actual signal will come between samples and thus go undetected. This probability must be faced under this operation. If the peaks are not sharp with respect to the sample rate or the sample rate is not a multiple of a periodic cause producing the peaks, the length of the interval of observation can be increased to make the probability of missing a peak (or perhaps 90 percent of its value) quite small. The sample time must, however, be traded off against the delay of making a decision – since decisions (based on the estimate of peak value over the period) can only be made at the end of the period. It should be emphasized that the output of this function represents an *estimate* of the true peak value. The validity of this estimate can only be determined from the statistics of the particular waveform.

Performance Criterion of Average Value

Implementation of this criterion to operate directly on the samples involves only minor changes from the integrator and low pass filter cases of Section 4.4.3.1. An integrator with a gain proportional to the duty cycle of the sampled pulses will be required and again, the integration must be performed over a fixed interval. Instead of dividing by the interval time, however, the result is divided by the number of samples observed. The greater the number of sample pulses observed, the less will be the variance of the estimate – with less and less responsiveness to changes. Variance for random samples is σ^2/N; where σ^2 is the variance of the original waveform and N is the number of samples in an observation. Sampled data pulses are rarely random since, if there were no correlation between pulses, the waveform could not be reconstructed. Correlation will increase the variance of the estimate.

Implementation of a low pass filter will involve special care in obtaining the true average since frequency and duration of the samples must be considered.

Performance Criterion of Integral Squared Error, Squared Error and Absolute Value

It is recommended that these criteria be implemented by digitizing and processing on a computer.

4.5 THE SMOOTHING OPERATION

BLOCK E.5

Section 4.4 described the design of hardware necessary to achieve a performance measure. Before this measure can be provided to the mapping operation, it will usually have to undergo smoothing to reduce extraneous fluctuations. The fluctuations or variability of the performance measure are the major sources of status errors. As such, the smoothing operation is the first place to look for error reduction.

It is significant to point out that reductions in variance are not achieved without penalty. The price to be paid is an increase in the time required to indicate a change in IBV status. *In general, reducing variance increases the time required to indicate a status change.* Achieving a balance between these two parameters is the most significant trade-off to be performed in design of the smoothing operation.

This section will describe design approaches to the smoothing operation and provide design aids to simplify implementation. Concepts of smoothing or time averaging have not been included. The reader wishing further information on this subject should consult Reference [2].

Implementation of the smoothing operation, like the implementation of special operations in Section 4.4.3, is sensitive to the type of signal being considered. As such, analog and sampled data signals will be treated separately. The former is addressed in Section 4.5.2 and the latter in Section 4.5.3. Digital signals can obviously be processed by the methods of Section 4.5.3. Section 4.5.1 addresses the nature of a typical signal to be processed by a smoothing operation.

4.5.1 Input Signal Character

A performance measure entering the smoothing operation can be envisioned as the following sum of signals.

$$s_c(t) = a(t) + n(t) + p(t) + d(t) \qquad (4.5.1)$$

where

$a(t)$ = trend component

$n(t)$ = zero mean white noise component

$p(t)$ = perturbation component

$d(t)$ = other dominant signals

The term a(t) is the quantity which is to be determined. The remaining terms constitute the fluctuations identified earlier. The trend component can behave in one of two ways.

a. a(t) can drift slowly as IBV parts age and environmental conditions change, finally exceeding a decision threshold. This represents a gradual, but not necessarily smooth, degradation. This behavior is the reason for the name, trend component.

b. a(t) can drastically change in a short time as a result of a catastrophic failure in the IBV.

The term n(t) represents the noise observed by the smoothing operation, the source of which is the IBV signal. This term is deliberately separated from the remaining two since its origin is different, its control is different and it will affect the estimating process differently.

The term p(t) is introduced to account for surges, spikes and dynamic responses to sudden inputs to, or loads on, the IBV. The source of these phenomena is the IBV signal. Such perturbations may not exhibit a zero mean value, even over extended periods of observation and will tend to bias the estimating procedure. They have been singled out in the equation for this reason.

The term d(t) is introduced to account for signals present in $s_c(t)$ which are not in the other terms. These would be signals that are of no immediate interest in determining status. One such signal will often be a hum.

The following two sections deal with implementation guidelines for determining a(t) using Equation (4.5.1) as a model of the input signal.

4.5.2 Analog Signals

Two types of devices are well suited for smoothing analog signals. These are the low pass filter and the constant time integrator. This section will apply general filter theory to achieve smoothing operation designs. Filter properties for assessing the relative virtues of one filter type over another are provided. The reader wishing more detail should consult References [3], [4] and [5].

Designing a smoother using low pass filtering is not a unique task. The problem is still one of rejecting energy outside the bandwidth of a(t) in Equation (4.5.1). The bandwidth of a(t) is determined by how rapidly a(t) varies, or equivalently, how rapid a rise one wished to detect. This characteristic is the response of the filter. It should be noted that, even if such a "filter" could be realized, one would not want to respond in zero time to a sudden change in a(t) since such a device would no longer smooth. It must then be admitted that a delay in the indication of a status change is inevitable. The designer would like to make the bandwidth of the filter large to realize rapid response. On the other hand, bandwidth must be reduced to reject noise and extraneous signals so that the estimate will have a small variance. (This corresponds to increasing the time effect of the filter impulse response in the time domain.) In the final analysis, variance of the estimate will be determined by the energy residue of n(t), p(t) and d(t) remaining in the bandpass of the filter. The term d(t) is often the biggest problem. If it contains no spectral components within the filter bandpass, all is well. On the other hand, if a dominant signal has significant energy at very low frequencies, the bandpass of the filter may have to be made quite narrow and quite sharp. This will result in sluggish response and is usually costly to implement.

Selection of a filter design for a given $s_c(t)$ cannot be made without knowledge of the characteristics of interest and how these characteristics apply to the problem at hand. For these reasons, several pertinent characteristics are discussed.

Noise Equivalent Bandwidth

Noise equivalent bandwidth is a measure of the capability of a filter to reject white noise. The filter noise power output with white noise as its input is proportional to the noise equivalent bandwidth.* It is therefore desirable to minimize this characteristic if $s_c(t)$ consists of a slowly drifting a(t) while p(t) and d(t) are negligible. This characteristic will be of interest in handling n(t).

---

*This description corresponds to single sided noise equivalent bandwidth which is used with single sided (i.e., only positive frequencies) noise spectra. Single sided spectra will be used throughout this section.

Power Spectrum (Spectral Response)

Power Spectrum is the ratio of output power to input power of a sine wave at a specified frequency relative to the maximum of this same ratio (in this case at dc), usually expressed in dB. The power spectrum can be used to determine the amount of rejection experienced by single, unwanted tones or narrowband interference centered at a given frequency. This characteristic will be of interest in handling d(t).

Impulse Response

Impulse response will give an indication of the filter's rejection of spikes. This characteristic will be of interest in handling p(t). Also, it may be very useful for comparing the hardware filter designs here with the computer implemented designs of Section 4.5.3 below.

Step Response

Step response is the time response of the filter to a unit step input. Of particular interest is rise time, the time from initiation of the step input to the point where output reaches 90 percent of its final value. Rise time is a measure of the delay of the filter in following discrete changes in the input and is of value in assessing notification time performance.

Four types of filters are evaluated for each of the characteristics identified above. These are the single pole filter, the double pole filter, the constant time integrator and, for purposes of comparison, the ideal rectangular filter. Except for the ideal filter, these filters were selected on the basis of their wide application and mathematical tractability of analysis. Multipole filters have by no means been ruled out nor have any of the active filtering devices. It should be recalled, however, that the designer is strongly motivated to realize the status development function simply and reliably.

The filter evaluations appear in Appendix B which develops mathematical expressions for the pertinent characteristics of each filter and provides selected application notes. Normalized plots of the evaluation results are shown in Figures 4.5.2−1 through 4.5.2−3. All plots involve noise equivalent bandwidth to facilitate direct comparisons. The following symbols have been used.

I_{max} = maximum value achieved by the unit impulse response (volts)

B_{eq} = noise equivalent bandwidth (Hz)

$R(t)$ = step response (volts)

t_r = time required for the filter output to reach 90 percent of step input (seconds)

T = fixed integration interval (seconds)

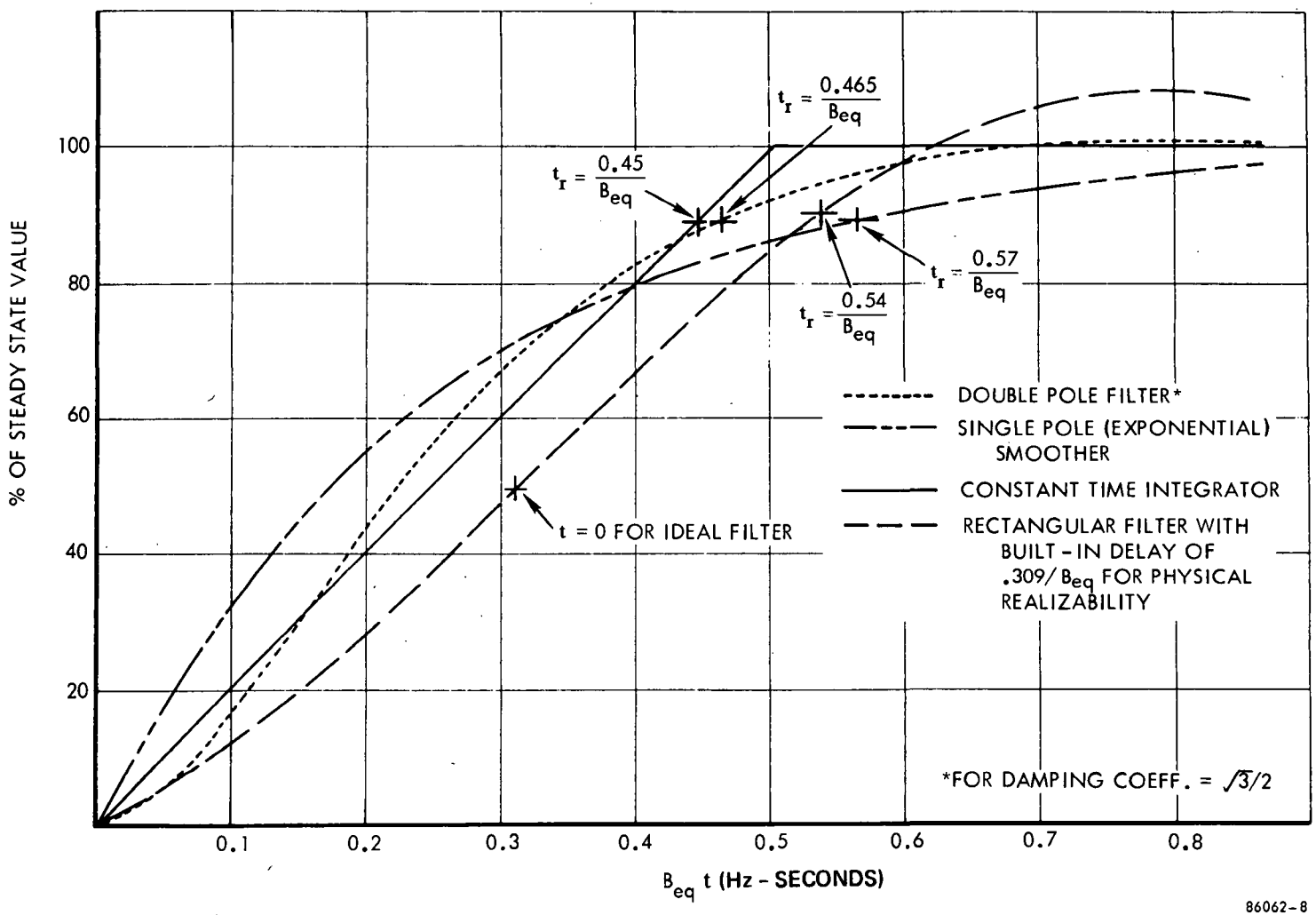


Figure 4.5.2–1. Time Response to Step Input

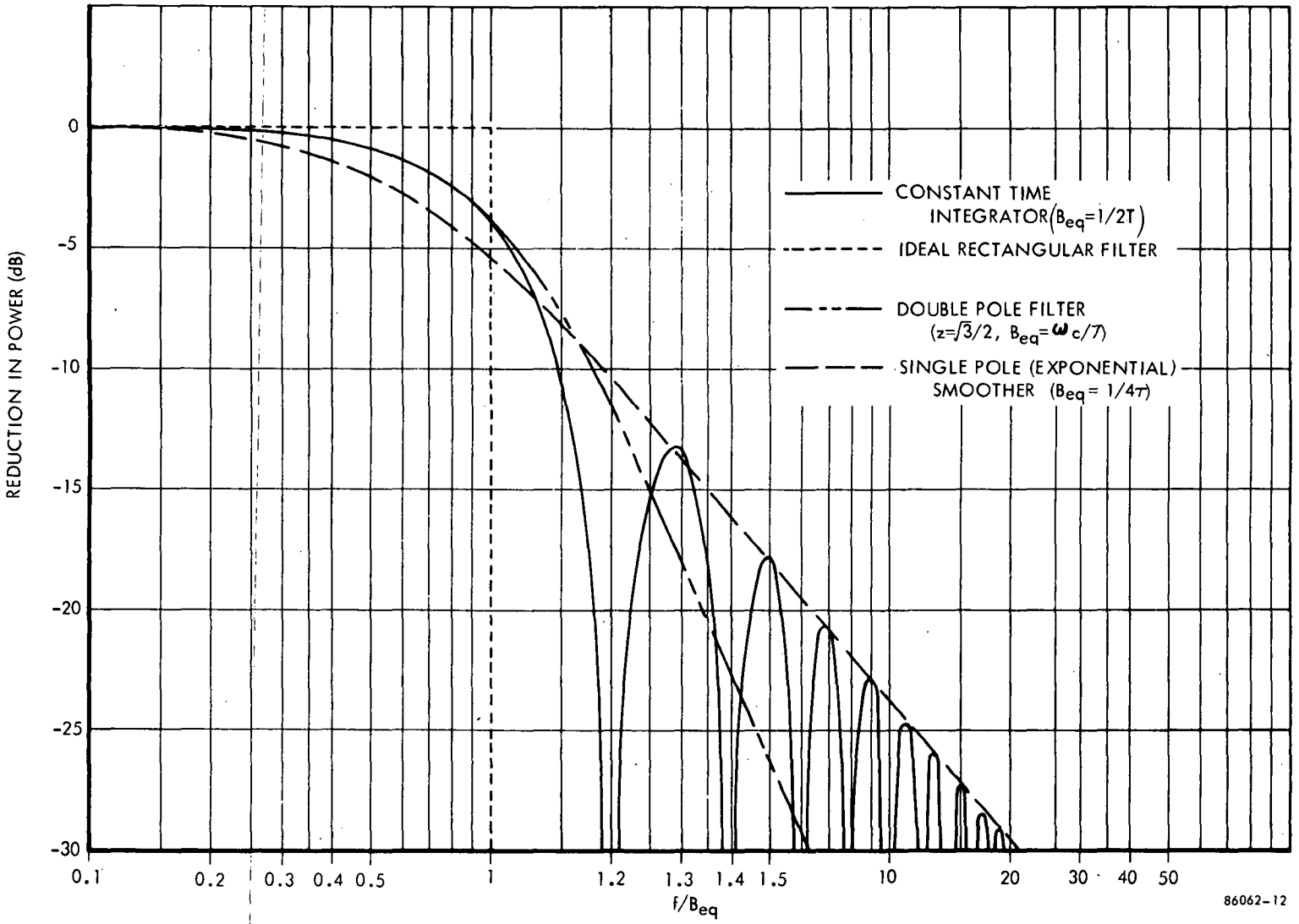


Figure 4.5.2–2. Spectral Response of Various Filters

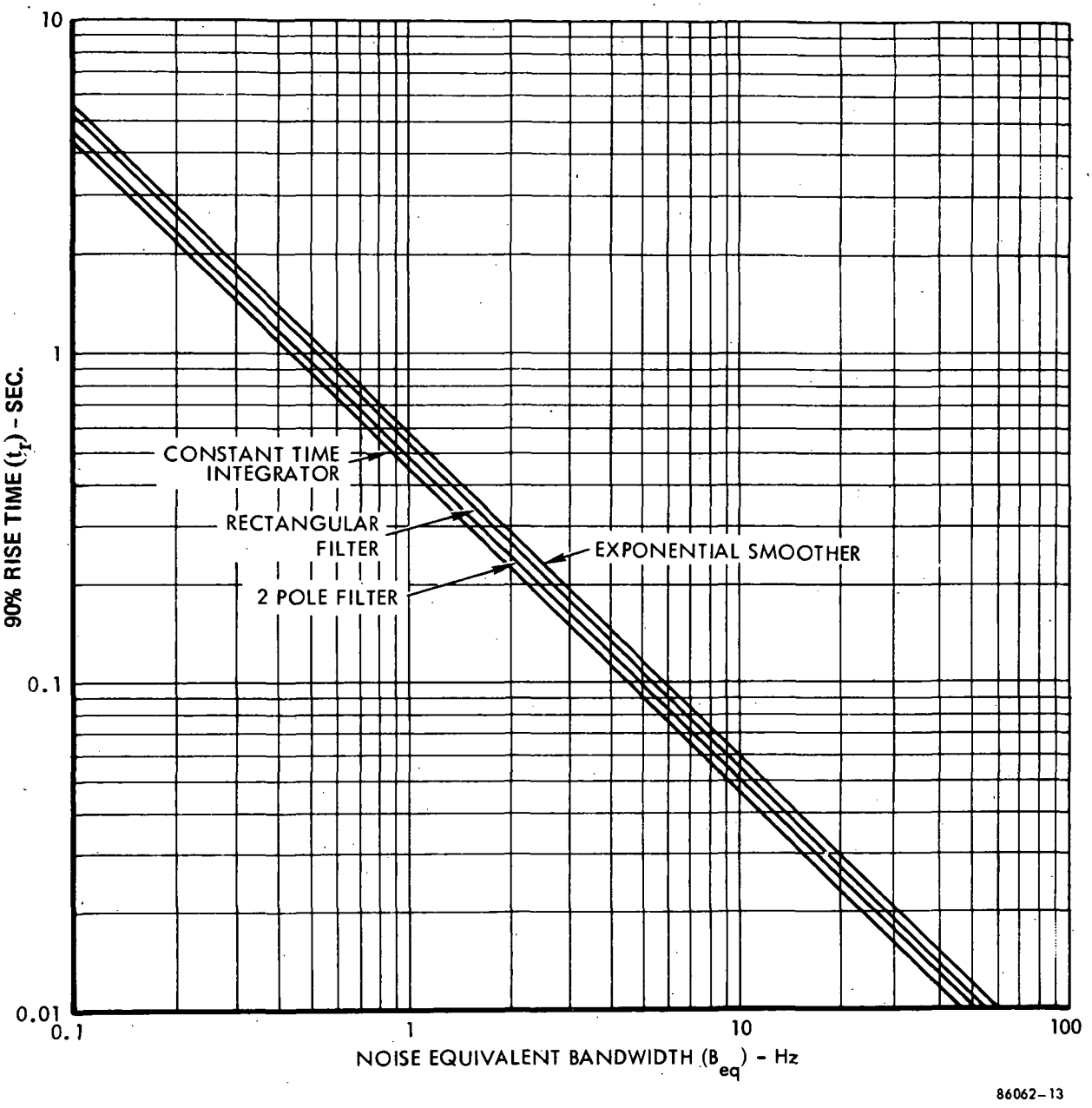


Figure 4.5.2–3. Rise Time Vs. Noise Equivalent Bandwidth

Table 4.5.2 summarizes the evaluations by comparing rise time, maximum inpulse response and power spectrum for each filter. The single pole filter has, in general, the poorest characteristics for use in the smoothing operation, but it is the easiest and least expensive to implement. The simple single pole filter should be entirely adequate for many smoothing applications.

The two pole filter is only slightly harder and more expensive to build unless active devices are to be used. This filter offers considerable improvement over the single pole filter in both spike rejection and tone rejection for the same B_{eq} or the same rise time.

The constant time integrator is somewhat more difficult to implement, principally due to the requirement for timing. The results are time sampled instead of continuous. If a sampled output and an increase in complexity is warranted, the integrator possesses the best rise time for a given noise equivalent bandwidth, the lowest impulse response and, as can be seen from the spectrum plot, convenient valleys in which known unwanted tones can be placed for rejection. From Appendix B, the integration time, T, is given as

$$T = \frac{1}{2B_{eq}} .$$

Table 4.5.2. Comparison of Filter Characteristics

Filter	Rise Time x B_{eq} ($t_r \cdot B_{eq}$)	Peak Impulse Response (I_{max}/B_{eq})	Spectral Response Power Vs. (f/B_{eq})
(1)	(2)	(3)	(4)
Single Pole	0.572	4.0	Relatively slowly decreasing with frequency
Two Pole	0.465	2.826	More rapid decrease – better tone reject
Integrator	0.45	2.0	Peaks slowly decrease – valley can reject known tones
Rectangular	0.54	2.0	Maximum tone rejection

The ideal rectangular filter is impossible to implement and can at best be approximated with difficulty. Because of complete rejection of higher frequencies, its impulse response is as low as that of the integrator. However, its rise time is, for the same reason, less than that of the integrator. The main reason for resorting to a filter approximating the rectangular would be a very high degree of interference at frequencies just outside of 1/(the desired rise time). It should be noted that, to achieve a reasonable approximation to the rectangular filter which is physically realizable, the response time plot of this filter has been delayed by $t = 0.309/B_{eq}$ in Figure 4.5.2–1. This time delay corresponds to the last zero crossing of the ideal waveform in negative time. The resulting rise time is $t_r = 0.54/B_{eq}$.

Implication of the above results and the use of Figures 4.5.2–1 through 4.5.2–3 can best be illustrated with an example.

Example

Suppose a smoothing operation must perform as follows. The output of the smoother is to be compared to a 1 volt threshold. It is desired that the effects of fluctuations about a(t) be less than 1/10 this value or 0.1 volt. If a threshold crossing does occur, it must be detected in one second for safety reasons.

$s_c(t)$ consists of the following worst case components: 40 volt spikes that last up to 1 ms, a hum of 0.5V peak at 1 Hz, a wandering hum due to beating of between 3 and 20 Hz with 0.1V peak, and a random white noise level of 2 mV^2/Hz. Comparing this problem statement to Equation (4.5.1), the spikes are represented by p(t), the white noise by n(t) and the two hums by d(t).

Since a threshold crossing by $\hat{a}(t)$ must be detected in one second, t_r is equal to one second max*. The minimum bandwidth that can be tolerated for this rise time is 0.45/1 = 0.45 Hz**. The 0.5 volt, 1 Hz hum is a potential problem. The peak amplitude must be reduced to below the 0.1 volt minimum, say by a factor of 10 (or 20 dB). A glance at the spectral response curves (Figure 4.5.2–2) shows the only filters that are down 20 dB by the time $f/B_{eq} = 1/0.45 = 2.2$ are the ideal filter and the integrator. The selection has been narrowed to one of these two. The integrator will be investigated first since this device is preferable if it can be used.

It is desirable to place the 1 Hz hum directly over a valley in the spectral response. Since B_{eq} must be greater than 0.45 Hz, the only place this can happen is for $1/B_{eq} = 2.0$ or $B_{eq} = 0.5$ Hz.

Using the value of 0.45 from column (2) of Table 4.5.2 ($t_r B_{eq}$ for the integrator), the rise time then becomes

$$t_r = \frac{0.45 \text{ Hz-Sec.}}{0.5 \text{ Hz}} = 0.9 \text{ seconds.}$$

---

* $\hat{a}(t)$ is the estimate of a(t)

**This quantity was obtained from Table 4.5.2 using the minimum value in column (2). Identical results could have been obtained from Figure 4.5.2–3.

The 40 volt, 1 ms spikes represent an impulse of 0.04 volt-seconds. For the integrator, Appendix B shows that $I_{max} = 2\ B_{eq}$ for a unit impulse. The 0.04 volt-second impulse will then result in a maximum integrator output of

$$I(max) = 0.04 \times 2.0 \times 0.5 = 0.04 \text{ volts.}$$

The roving hum from 3 to 20 Hz gives values of f/B_{eq} of from 6 to 40. A glance at the spectral response (Figure 4.5.2–2) shows the minimum attenuation in this range as 20.5 dB. This reduces the 0.1 volt hum to 0.01 volts peak.

The white noise of 2 mV^2/Hz will produce an rms output given by the noise equivalent bandwidth as

$$(2\ mV^2/Hz \times 0.5\ Hz)^{1/2} = 1\ mV \text{ rms.}$$

The three sigma value which would constitute a typical constraint is 0.003 volts.

The total worst-case output deviation due to $s_c(t)$ is:

.040 – spikes
.010 – wandering hum – max
.003 – random white noise
.053 volts total

This value is well within the specified 0.1 volt limit. The constraining parameters on the filter are minimum rise time and rejection of the 1 Hz hum.

The integration time for this integrator is given by

$$T = 1/(2\ B_{eq}) = 1 \text{ second.}$$

Of the smoothing devices considered, the best choice for this hypothetical case is then a constant time integrator with an integration time of 1 second.

4.5.3 Sampled Data and Digital Signals – Computer Implementation

This section addresses a computer or software implementation for smoothing when digital data or sampled data constitute the performance measure. The technique discussed is known as recursive smoothing and has the advantages of, a) requiring very little computer storage, b) being readily implementable on a real time machine and c) providing an estimate at each sample time. Results from a smoothing function implemented in this manner will obviously be digital and the subsequent mapping operation can be readily implemented on the same computer.

The analogy between this implementation and those of Section 4.5.2 will become obvious as the description continues. The section treats first order or exponential recursive smoothing to relate the concepts and then treats the general case. For the reader wishing additional detail, Reference [6] is an exceptionally readable text on the subject of recursive smoothing.

4.5.3.1 First Order (Exponential) Recursive Smoothing

First order recursive smoothing is the sampled data equivalent of the single pole filter discussed in Section 4.5.2. The technique applies geometrically related weights to successive time samples to achieve the estimate very similar to the impulse response of a filter. The expression describing the process is

$$A(kT) = \alpha x(kT) + (1-\alpha)\, A\,[(k-1)T],$$

$$0 \leq \alpha \leq 1$$

where T is the time between samples, x(kT) is the value of the $k^{\underline{th}}$ (present) sample and kT is the kth sample time. α is the smoothing constant. A(kT) is the estimate of the average based on all previous samples. All that need be stored for this process is A(kT). By going through a few steps of the indicated operation in the above equation, it is easy to see the geometric relationship from whence the process gets its name. The next step is selection of α. As indicated in Section 4.5, selection of α will be a compromise between variability and response. The weights applied to the samples are $\alpha(1-\alpha)^m$ where m represents the sample taken m periods ago. Large values of α will yield a responsive smoother and small values a more stable smoother. While it will be extremely difficult to determine variance of the estimate for an arbitrary S_c (kT), the discrete equivalent of s_c (t), one method of rating values of α is to consider a(kT) constant (see Equation 4.5.1) and disturbed by a process with zero mean. This restriction is not as severe as it first may seem since a slowly varying a(kT) could be considered constant over the effect of the filter. The variance for uncorrelated samples is

$$\sigma_e^2 = \frac{\alpha}{2-\alpha}\,\sigma_s^2 \qquad (4.5.3-1)$$

where σ_s^2 is the variance of $S_c(kT)$ and σ_e^2 is the variance of the estimate. A plot of σ_e^2/σ_s^2 versus α is shown in Figure 4.5.3.1. In this figure, samples that are correlated will have a greater σ_e^2/σ_s^2 than random samples (but still less than the variance of S_c (kT)). It is safe to assume that, under the conditions cited for Equation (4.5.3–1), correlated samples will have, for the same value of α, values above the plotted curve. Response, or the number of steps required for a first order smoother to reach 90 percent of the height (final value) of a step input, is given by

$$0.9 = 1-(1-\alpha)^{n+1}$$

$$n = \frac{1}{\log_{10}(1/1-\alpha)} - 1. \qquad (4.5.3-2)$$

A plot of Equation (4.5.3–2) is also shown in Figure 4.5.3.1. Looking at Equations (4.5.3–1) and (4.5.3–2), it is possible to infer the behavior of the smoother as a function of the smoothing constant, α.

Applications of first order recursive smoothing are found in digital process control and in the field of business trend analysis. While application is expected in implementations of status development, the response versus variability characteristics are a limiting feature. The following section will address higher order smoothing techniques.

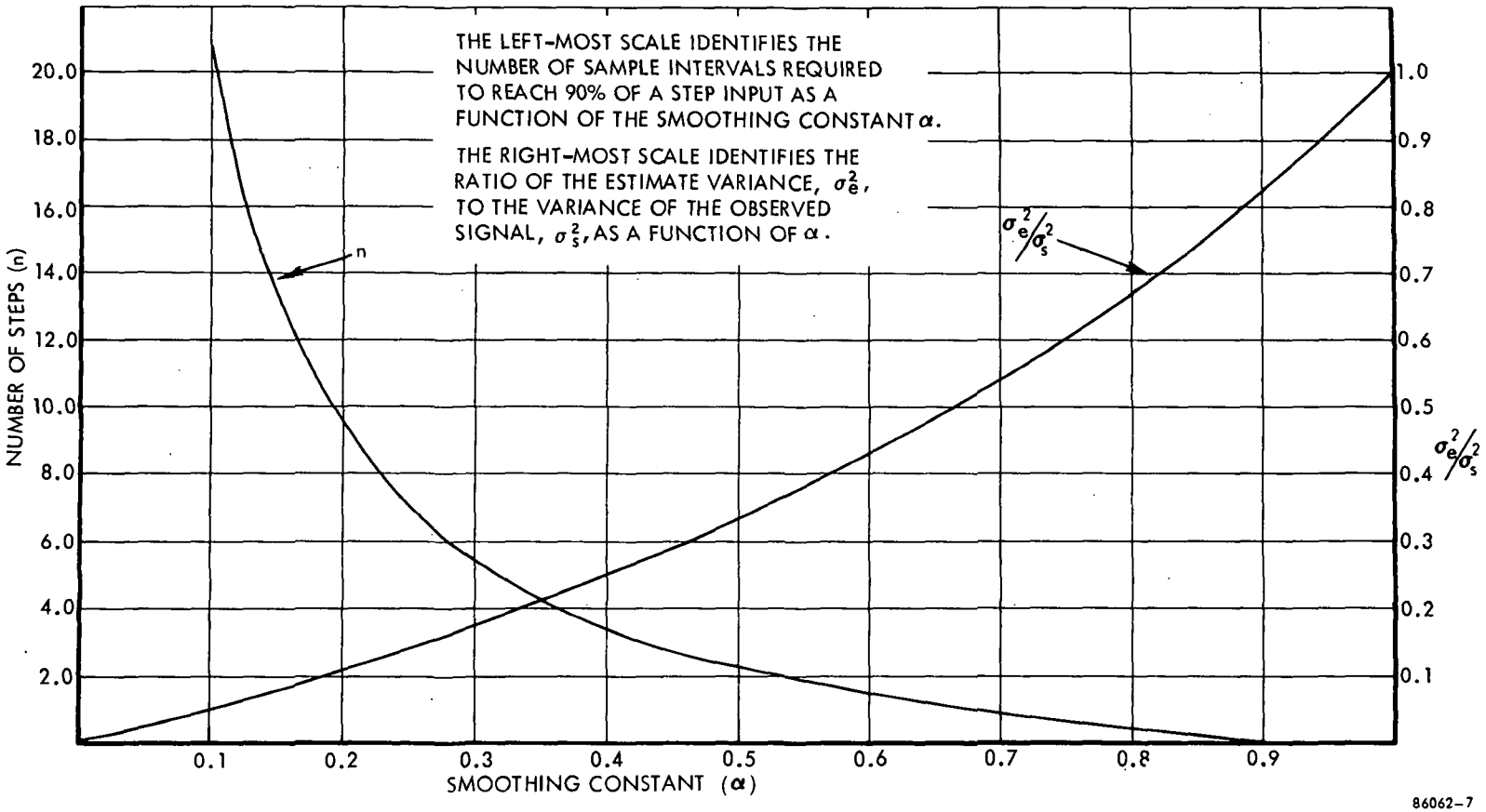


Figure 4.5.3.1. Performance Characteristics of a First Order (Exponential) Smoother

4.5.3.2 Higher Order Recursive Smoothing

Just as higher order filters (increased number of poles) were used in Section 4.5.2 to achieve equivalent response with a smaller bandpass, higher order smoothing may be used to accomplish the same end. In general, n-order smoothing is realizable but interest here will be limited to no greater than third order. There is a great deal of correspondence between the characteristics and responses described in this section and those of Section 4.5.2. It is not the intent to repeat previous material. Attention will be focused on establishing smoothing characteristics and relating these results to those of Section 4.5.2.

Second order smoothing essentially imbeds the first order process in an identical but overlying process. Operationally, second order smoothing determines the first order estimate and uses this estimate (as opposed to the observed value) to update the second order estimate. Then

$$A_2(kT) = \alpha A(kT) + (1 - \alpha) A_2 [(k - 1)T]$$

where $A_2(kT)$ is the second order estimate at time kT (or period k) and A(kT) is the first order estimate. This technique will require two locations in memory since both $A_2(kT)$ and A(kT) must be preserved for the next period. (Recall that α must either be stored explicitly or entered as a constant in the program.)

Third order smoothing continues the trend set by second order and operates on the second order estimate.

$$A_3(KT) = \alpha A_2(KT) + (1 - \alpha) A_3 [(K - 1)T]$$

As might be expected, third order smoothing requires three memory locations.

A comparison should be made between the three smoothing techniques with regard to response. The basis for comparison is response to a step input. If α_1 is the smoothing constant for first order smoothing, α_2 second order smoothing, etc., an equivalency among these constants can be established for response to a step input. This equivalency is shown in Figure 4.5.3.2. From this figure, it can be seen that, for an equivalent response to a step input, a second order smoother would require an α of 0.37 and a first order smoother an α of 0.6. These results can be related to the response plot of the first order smoother in Figure 4.5.3.1. For $\alpha = 0.6$, a first order smoother would reach 90 percent of its final value in two steps. A second order smoother would achieve the same result with an $\alpha = 0.37$. While the variance curve for second order smoothing is not shown in Figure 4.5.3.1, it will exhibit the same trend as that shown for first order. It is obvious that a reduction in estimate variance (with respect to the variance of the observed signal) is achieved.

Some further characteristics are summarized in Table 4.5.3.2. The impulse response is valuable for determining rejection of spikes and providing comparisons to the results of Section 4.5.2. The transfer function will afford z-plane analysis.

It is helpful to draw analogies to the characteristics identified in Section 4.5.2. The ratio of σ_e^2/σ_s^2 in Figure 4.5.3.1 is directly analogous to noise equivalent bandwidth. The relationship between the impulse responses of Table 4.5.3.2 and a filter impulse

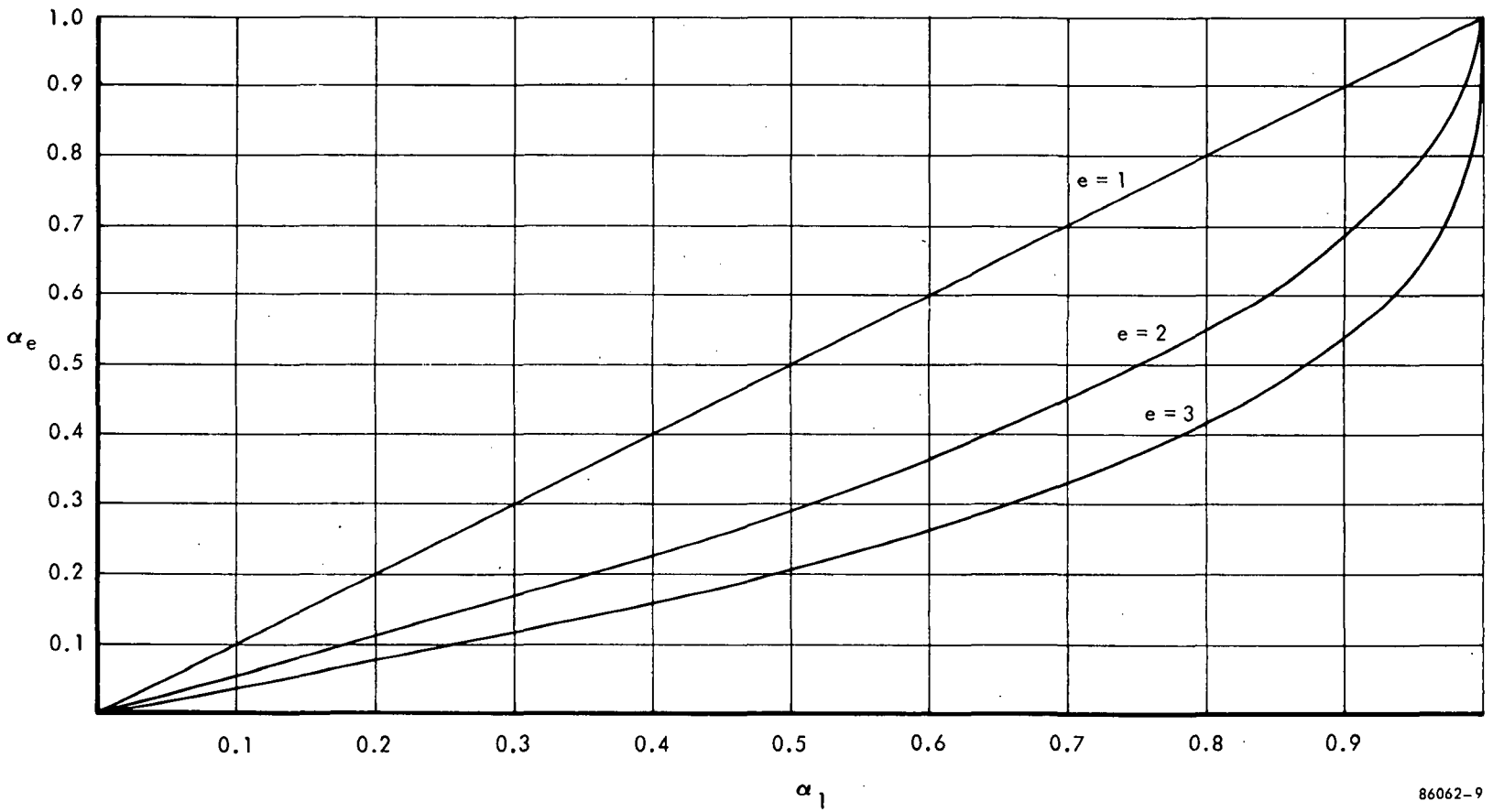


Figure 4.5.3.2. Equivalency of Smoothing Constants for Three Orders of Recursive Smoothing

response is obvious. The characteristic of step response between the two approaches is also straightforward. Figure 4.5.2–3 can be used to obtain rough frequency domain information on first and second order smoothing. This is accomplished by determining the rise time from Figure 4.5.3.1 and consulting the curves for the exponential and double pole filter in Figure 4.5.2–3.

Table 4.5.3.2. Some Characteristics of the First Three Orders of Recursive Smoothing

Order of Smoother	Impulse Response	Transfer Function	Req'd. No. of Stored Data Words
First	$\alpha(1-\alpha)^t$	$\frac{\alpha}{1-(1-\alpha)z}$	1
Second	$\alpha^2(t+1)(1-\alpha)^t$	$\frac{\alpha^2}{[1-(1-\alpha)z]^2}$	2
Third	$\alpha^3 \frac{(t+1)(t+2)}{2}(1-\alpha)^t$	$\frac{\alpha^3}{[1-(1-\alpha)z]^3}$	3

4.6 DECISION RULES AND MAPPING

BLOCK E.6

Having identified a performance criterion and provided the wherewithal to transform this criterion into a status variable, it will now be necessary to establish decision rules for the criterion and implement these rules into the mapping operation. The designer must determine the number of indications to be used for status, i.e., the number of status levels, and identify a decision rule which divides each indication.* Then, if the number of status indications is two (e.g., go/no-go), a decision rule must be determined which divides these two indications. Once the decision rule has been identified, it must be implemented into the mapping operation in the form of a value threshold so that equivalent decisions can be made about the status variable. Recall from Section 4.3 that *mapping thresholds must be constant values and can operate only with deterministic status variables.*

It should be recalled that a single IBV will usually require several performance criteria to achieve a reasonable confidence in the status results, i.e., a multidimensional status variable. There will be, then, a mapping or threshold device for each criterion and the results of these individual devices must then be combined to form a composite statement of IBV status. The former device is called a first level mapper and the latter device, a second level mapper.

The first level mapping device may well contain more than just threshold circuits. Consider a decision rule that states a property is satisfactory if the status variable lies

*To increase information relayed to the equipment user by status development, the designer is often motivated to use three levels of status. In theory, this provides an indication of marginal conditions in an IBV. The approach, however, has a serious drawback when automated decisions are to be made regarding several independent status indications, each with three or more possible states. Such decisions are always encountered in status resolution and the process of combining status of several IBV properties into a composite IBV status statement.

between -2 and +4 volts and is unsatisfactory otherwise. Unfortunately, threshold circuits will only determine if inputs are above or below a single value. While this is a one dimensional case, two thresholds must be tested: implying two threshold devices. The results of these two devices must be combined to determine status of the single criteria. The combination will usually be implemented with logic devices.

In addition to the devices thus far identified, the mapping operation may also include interface and timing circuits. The interface circuitry may be used at the mapping operation input to perform level conversions, inversions and the like. The output interface might include digital encoders or level converters. If digital data, sampled data or *status holding* are involved, timing will be required.

Implementation of the first and second level mappers are discussed in Sections 4.6.1 and 4.6.2 respectively. Section 4.6.2 will also describe considerations of multidimensional status variables.

4.6.1 First Level Mapper

The principle of the first level mapper is not complicated and has been described in numerous places throughout the handbook. Of interest here will be its implementation. Just as signal types influenced implementation of the smoothing operation, so too, will they influence implementation of the first level mapper. Since the mapper is a threshold device, the distinction between analog and sampled data signals is not important. Consequently, the two types of signals can be lumped into a single class of analog signals. The threshold operation in both cases will be the same except that timing will be required for sampled data signals. When sampled data signals are being processed, however, status indications can only be realized while the data are present. As such, provisions must be made to hold the status at its last condition until new data arrives.

With the precautions identified above, the mapper can be considered to handle digital and analog signals. Digital signals will typically imply digital implementation of the mapper. If a general purpose digital computer is used to determine the status variable, it can also be used for mapping since decisions on relative magnitudes can easily be made with any computer.

If a digital computer is not used, a logical mapper is the obvious candidate to perform status evaluation. The logical combination of input states to produce the required status indications may be very simple or very complex depending on the encoded format representing these states. For instance, if the status variable is the output of a digital counter, it would be wise to design the counter to initialize at the threshold or decision boundary and decrement the count. At the end of the counting period, a simple operation of checking for ones would indicate status.

Such simple designs of digital mappers are not always possible. For instance they would have to be compromised if more than one change of status must be detected or if hardware is used which is not specifically designed for the task at hand.

For status mapping of analog signals, the range of the input must be partitioned into regions, each of which corresponds to a specific output state. Implementations of a mapper must consist of devices which can determine when the status variable possesses values which correspond to these ranges. Such devices are by definition, threshold devices

(e.g., Schmidt trigger, operational amplifier bang-bang). There are numerous devices meeting this description and numerous references available from which to select and design threshold circuits. It is not the intent to duplicate this information but to indicate requirements for the circuits. One such requirement is that the output voltages of the first level mapper must be compatible with demand of the second level mapper. It will be a rare case when ranges of status variable voltage match these demands. As a result, the threshold circuit will inevitably employ an active device. Active devices will also offer the high input impedance usually required of the smoothing operation. It should also be noted that the decision thresholds are implemented within the threshold circuits using voltage references such as Zener diodes and precision voltage dividers. *As a rule of thumb, the threshold stability and precision should be specified at least one order of magnitude less than the distance between thresholds, or the dynamic range of the status variable if there is but one threshold.*

4.6.2 Multidimensional Status Variables and the Second Level Mapper

In general, the status of a device will be deduced from more than one status variable. These variables may arise from a single signal with several properties which are indicative of operation or they may in fact come from several signals, each having properties which are indicative of operation. In either case, these status variables will be referred to collectively as a multidimensional status variable since they will affect a decision of status on the same IBV.

The general methodology for the mapping of a multidimensional status variable into conditional status is as follows.

a. On each dimension (property), decide conditional status as if this dimension were the only one present. This is the operation of the first level mapper as described previously. These decisions should be mutually independent, that is, the outcome of any one decision should not depend upon the outcome of any other decision.

b. Perform any necessary level conversions on the output of the first level mapper to achieve compatibility with subsequent logic (for each dimension). It will then be convenient to describe a "1" as indicative of an acceptable operating state for a dimension and "0" for unacceptable operation for a dimension. (This step points clearly to the desirability of 2-level status as indicated in Section 4.6.)

c. Prepare a truth table which relates the combined states of each dimension to conditional status for the IBV. This table will frequently establish the following logical operation.

   - All dimensions contain "1", implies IBV status acceptable
   - IBV status unacceptable otherwise

d. Implement the truth table into logic. The logical operation in (c) above is a simple AND gate.

To gain insight into how the method works, consider a three dimensional status variable where the dimensions describe amplitude, frequency, and phase of an IBV output signal.

By some *a priori* knowledge, the acceptable values of the three parameters, amplitude, frequency, and phase, are known independently of each other, and therefore a status decision on each can be made without affecting or influencing a similar decision on the others.

It is also known that the principal system constraints are such that the system will operate if all or any two of the three IBV parameters are within specification. For example, the device may be a phase locked loop and while improper amplitude may be undesirable, if frequency and phase are valid, the automatic gain circuits in the following equipment may compensate for this fact. Similar arguments for the frequency and phase parameters may be introduced by noting that, if in fact the IBV is a phase locked loop, such devices require time to "lock up." By letting an acceptable variable be denoted by a one and an unacceptable variable by a zero, the associated truth table for IBV conditional status is:

Dimension Status			Cond. IBV Status
Amp.	Freq.	Phase	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

The truth table then indicates the implementation of the second level mapper.

Implementation considerations would hardly be complete without addressing the dimension effects on mapping errors. How many status variables are required to represent IBV status? Aside from adding equipment, can this number be made as large as desired? Obviously, there must be some fixed number of dimensions such that if more dimensions are used, the increase in knowledge of the IBV's status is offset by either an increase in probability of error, by the increasing complexity of the circuits needed to realize the decision function, or by both.

Consider the case where the probability of making an error in the status of each dimension is a constant p and there are M dimensions. For sake of simplicity,

assume the logic which maps the first level status into conditional IBV status is error free. The distribution of the number of errors will be binomial. For example, suppose M = 10 and p = 0.1. Then 65 percent of the time one would expect to see an error. If these ten dimensions were "concrete" specifications, all of which absolutely had to be met, IBV status would be in error more than half the time due only to chance. This could be disastrous and readily exemplifies the impact of added dimensions.

4.7 INTEGRATION OF DESIGN

BLOCK E.7

Sections 4.4 through 4.6 have addressed the detailed segments of status development design. These segments must be integrated into a complete structure which extracts IBV signal properties and provides status indications on an automated basis. This section offers the first opportunity to view status development as a whole. Overviews of this nature are best presented by way of example. The material that follows discusses five sample implementations which have been selected to portray a particular facet of the overall design.

Implementation A

Figure 4.7–1 depicts an implementation using integral squared error (ISE) as a special operation to form the performance measure. The reference is formed by comparing the output amplitudes of two identical, redundant elements. (The outputs are assumed to be continuous.) The detection/measuring devices could be transducers, filters (to eliminate spurious signals which are of no consequence to the performance measure), amplifiers, shapers and/or simple isolation mechanisms. A balance has been introduced to account for any phase differences between the detecting/measuring devices as well as cable routing. The balance will also account for peculiarities of different elements when element replacement is necessary during maintenance. Since integration is being performed on a wholly positive quantity (squared error), a timing circuit will be required to control the interval of integration.

The implementation does not use a smoothing operation, therefore, the output of the ISE device is the performance measure and the status variable. Since a single property was selected to indicate operation, there is but one status variable and the mapping operation is a simple threshold device to implement the decision rule. Output of the mapping operation can be an n-level signal representing status of the IBV. In this implementation, however, the IBV is actually the redundant set and the only status statements which can be made are: (a) both elements operative, (b) one element operative, other element inoperative. This is obviously a 2-level case. Note that both elements would have to fail at the same time for a third condition of "two inoperative elements" to exist. If there is a reasonable likelihood of this event occurring, the use of compare-two as a reference technique should be seriously questioned. This comes about since compare-two, using no further information, cannot detect two simultaneously inoperative conditions.

It should be noted that, due to the integration interval, the decision rule can only be stated in terms of the value of the status variable at the end of this interval. This implies that, if the integration interval is T seconds long, status indications can only be achieved every T seconds. The status variable here represents samples of ISE values. Since smoothing is not used, it is desirable that the integration period be long enough to collect a "good" sample and at the same time not be so great as to adversely affect reaction time to changes in IBV status.

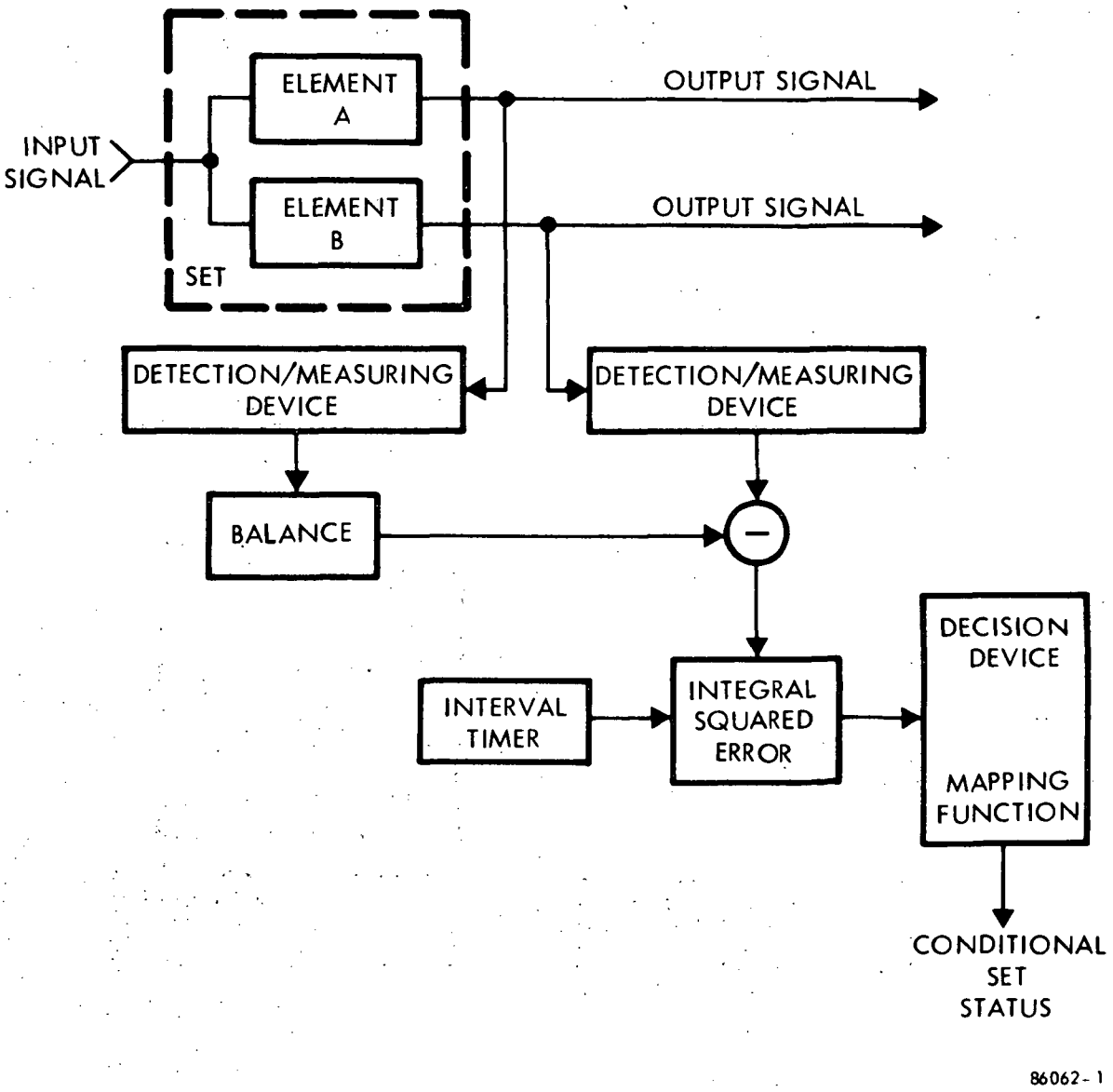


Figure 4.7—1. Implementation A

Implementation B

Figure 4.7–2 depicts an implementation for a digital IBV (single element) using bit error rate as a special operation to form the performance measure. The reference is formed by providing the inverse of the IBV transform and comparing this result with the IBV input. Since there are two outputs, the implementation has indicated the general inverse transform flow to arrive at the input signal. For example, the IBV might first be performing a level transformation on the input and then modulating two different signal sets. A delay has been incorporated to time balance the inputs to the inverse transform of function C. An additional balance and delay has been provided to achieve coincidence at the difference point.

Errors are accumulated over a time base to provide an estimate of the bit error rate. This estimate forms the status variable (again, smoothing is not used) and status information is available only at the ends of the period established as the time base. It is important to note that, presumably, parity is not included at the IBV input or else inverse transform would probably not be required.

In theory, there can be several levels of status indicated by the mapping operation, e.g., green, amber, red. The wisdom of this is questionable, however, without the use of smoothing to reduce the variance of the status variable. *It is also important to note that, while a status of "amber" is meaningful to an individual, its practicality is limited for automated status resolution or automated fault recovery systems (see footnote, Section 4.6).*

Implementation C

Figure 4.7–3 depicts an implementation using two performance criteria, each employing a different reference scheme. The first criterion is a measure of the frequency difference between input and output of the IBV and, as indicated by the criterion, the reference is formed by inverse transform of this property. The discriminator provides a voltage analog of this difference which is the performance measure. Since smoothing is not used, this variable is also the status variable used by the first level mapper to determine status of this property.

The second criteria is a typical implementation of the property of overshoot or noise spikes. It may be that the device receiving this IBV output will critically saturate or heat up with high frequency noise spikes of extended duration. It may also be known that such spikes can occur if the IBV is not performing properly. The performance criterion would then be concerned with the average energy, or more conveniently, voltage-time above some threshold. The clipper establishes this threshold and all voltage amplitudes above this threshold are integrated over a time base. In this case, then, two special operations are performed. Output of the integrator is gated (not shown) to the low pass filter at the end of the integrating time. The low pass filter smooths the estimates and performs the smoothing operation. The results are then passed to the mapping operation for this property. Since the input to the mapper is average volts-seconds and not energy, the decision rule must be adjusted accordingly. Note that the reference for this property is established in an absolute sense.

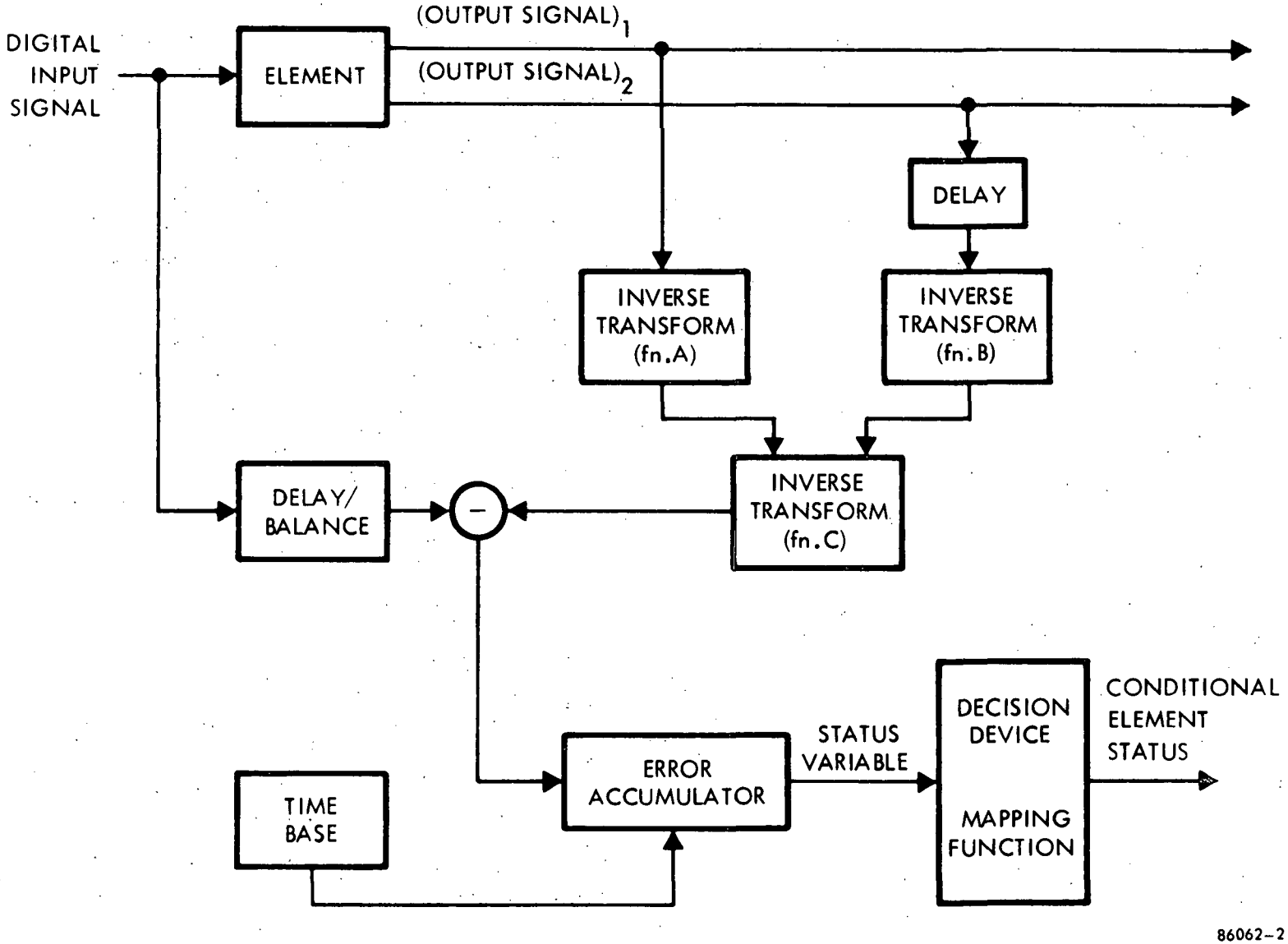


Figure 4.7–2. Implementation B

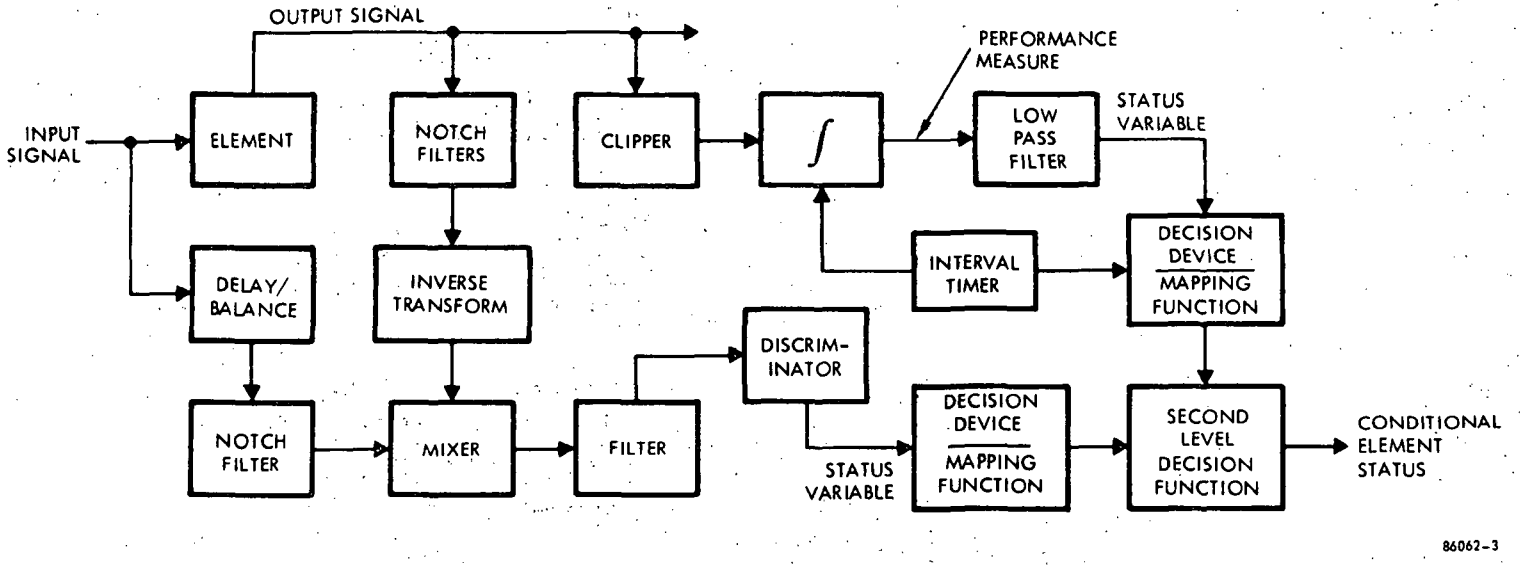


Figure 4.7–3. Implementation C

The implementation involves a two-dimensional status variable. Consequently, status of each property must then be mapped into status of the IBV. This is performed by the second level mapper.

Implementation D

Figure 4.7–4 depicts an implementation using two performance criteria. The criteria are concerned with a single property in each of two output signals from an IBV. The reference for each is established in an "absolute" sense but the references are time variable. As such, it is not convenient to use a threshold in the mapping operation for the reference (as has been done in previous examples using absolute references).

The top part of the implementation operates on the amplitude of output signal number one and compares this with a time function which represents the true or normal behavior of this amplitude. One is not always concerned with the total departure of a function from the norm, but often with the percent departure. This is implemented with a voltage divider. The performance criterion would then be concerned with the percent departure of signal amplitude from the normal or reference function and a decision rule for this criterion would then be implemented into the mapper for this property.

Output of the percent difference calculator (which is a special operation) is the performance measure. This measure is then smoothed in the integrator (smoothing operation) to form the status variable. One application of this criterion might be the x-coordinate of a missile trajectory. The expected trajectory forms the reference and the actual trajectory would be signal number one.

The lower part of the implementation also employs a time variable reference but does not use percent error as a criterion. Smoothing is also not used. This property could be a well behaved function such that smoothing was not necessary. Confidence statements might then be made about the total, unsmoothed comparison such that a decision rule can be formulated for inclusion into the mapper.

Implementation E

Figure 4.7–5 depicts an implementation using two properties of one signal and a single property of a second signal to form a 3-dimensional status variable. Both signals are outputs of the IBV. Status variable 1A is a measure of a performance criterion with a reference established in the absolute sense. This allows the decision rule to be incorporated into the mapper threshold directly. Status variable 1B is a smoothed version of the property of interest and it too has a reference in the absolute sense.

The lower portion of the implementation calculates the average value of a property. Thus, the performance criterion is concerned with the average value of this property and smoothing is not used. The gated input to this portion of the implementation is driven by a random noise generator. This provides random samples of the signal property. These samples will approach the classic notion of random samples if the range of the time between samples is at least ten times the value of the point where the signal autocorrelation function approaches zero and time between samples is uniformly distributed.

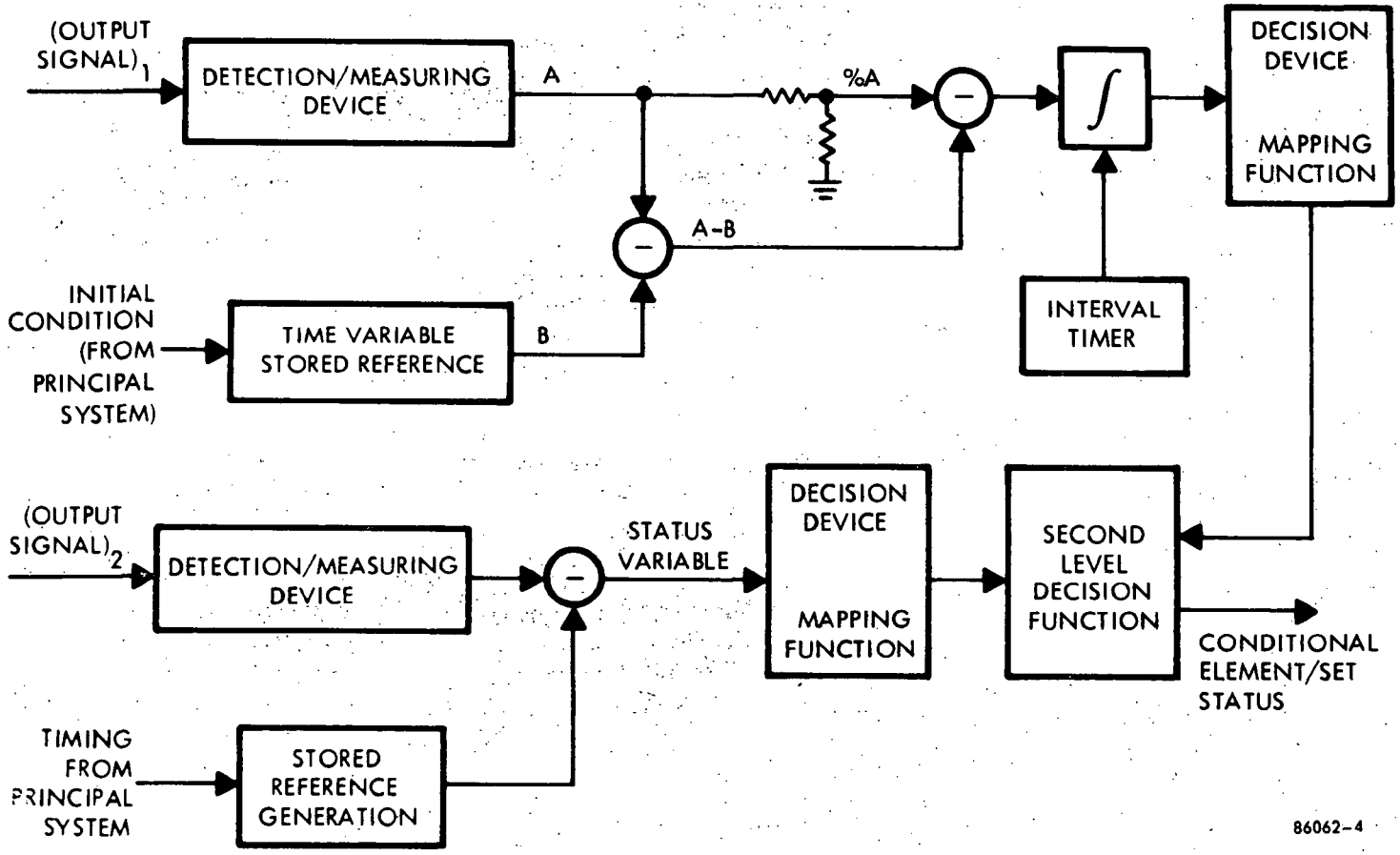


Figure 4.7–4. Implementation D

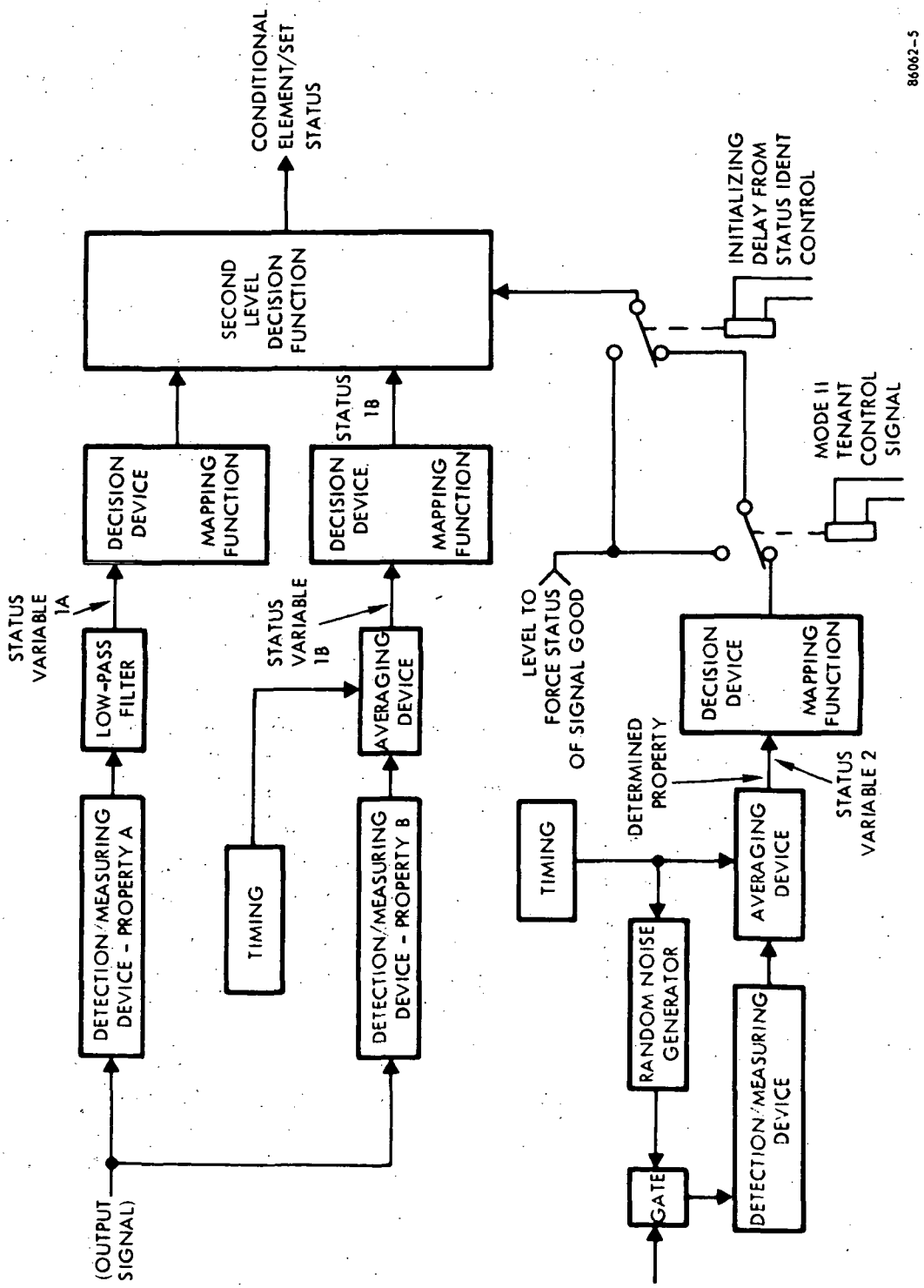


Figure 4.7–5. Implementation E

Control relays have been introduced into this implementation to emphasize initialization and operating mode difficulties. Averages take time to calculate since samples must be collected. When an IBV is first energized, there may be an erroneous output of the property (first level) mapper until operation stabilizes.* A time delay relay would solve this problem by forcing a voltage into the second level mapper which corresponds to satisfactory operation. The second relay is involved with operating modes of the IBV. It may be that under one of these modes, output signal number two is not used. Any indications of status from this signal would be meaningless and very possibly misleading -- especially if the decision rule stated the amplitude must be greater than some minimum value. Under these conditions, status based on indications from this signal would again be forced as described above.

4.8 ERROR ANALYSIS

BLOCK E.8

The topic for discussion in the ensuing paragraphs is errors; their sources and techniques for their minimization. At first, it will be convenient to think of errors only as mistakes, however, a more complete and limiting definition of errors will be advanced in a later paragraph. It should be pointed out at the outset that a clear distinction has been made between failures in the status development function and errors committed by the same. The former is a nonrecoverable condition and its causes and control are different from those of errors. Errors represent a recoverable condition and are committed by chance circumstances within the function.

The status development function commits an error whenever status is erroneously indicated. The seriousness of this error, and consequently the amount of effort which can justifiably be expended on an error analysis, depends on how the status indication is used. If the error simply causes a light to flicker on a console; that is one thing. If the error causes a mission abort or initiates a sequence of automatic reconfigurations; this is quite another prospect.

Errors originate from three major sources:

- circuit considerations within the status development device,
- variations and perturbations in the status variable, and
- setting of mapping thresholds.

Errors arising out of circuit considerations can be directly traced to four causes.

- Accuracy and drifts in power sources
- Drifts in timing and time delays

*This could be quite critical if the results of status are used for automatic fault recovery.

- Accuracy and drifts in piece parts

  resistance voltage dividers

  Zener voltages

  R/C timing

  filter skirts

- Noise, self-induced and received

The mechanisms behind part parameters drifts are temperature and aging. Control of these drifts and requirements for part accuracy are problems in statistical circuit design. These design techniques are well covered in the literature and will not be pursued here. The time honored techniques of noise control – limitation of bandwidth, reduction of part operating temperature and shielding – apply equally well to status development circuitry.

In summary, circuit considerations are under the designer's control. The overriding concept is to limit errors caused by circuit considerations to the extent that they are small compared to errors originating from the variation of the IBV signal.

Errors due to variations of the status variable and setting of mapping thresholds are closely related. These error sources are independent of circuit considerations and will be described assuming circuit errors are nonexistent.

There are two types of errors which can be made. One occurs when an acceptable IBV is identified as being unacceptable. The other occurs when an unacceptable IBV is not identified as such. Both errors are of interest over periods of time. The probability of committing the types of errors are then defined as:

Pr(Type I)	=	probability that an acceptable IBV is identified as unacceptable over the period of its mission
Pr(Type II)	=	probability that an unacceptable IBV is not identified as such during the notification time

The cause of these errors can be viewed conceptually as shown in Figure 4.8. In the figure, the trend component of the status variable (see Section 4.5) is shown crossing the mapping threshold as the IBV fails. Variability of the status variable is depicted by the distribution superimposed over the normal level. The shaded portions of the distributions indicate the Type I and II error probabilities. It should be noted that this illustration of voltage distribution does not, however, portray the complete picture. Recall that time entered into the failure definitions. Figure 4.8 does not explicitly account for time. As time increases, the probability of committing a Type I error increases. The figure is best viewed as the probability of error for a single decision. As more and more decisions are made, the probability of at least one decision being in error increases.

[4.8.1]

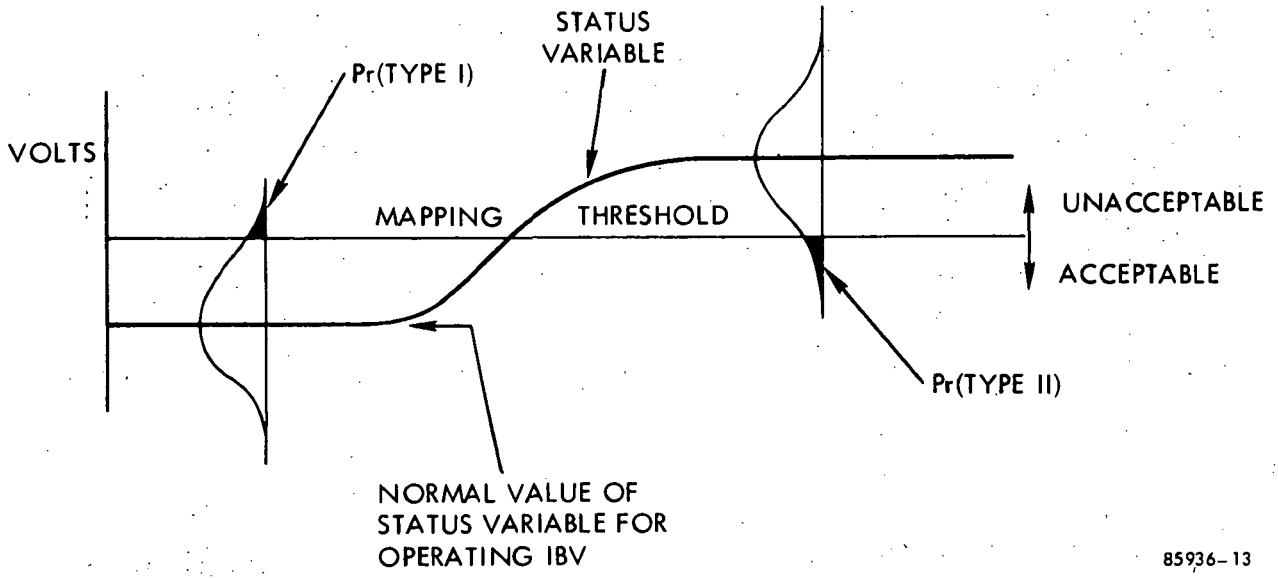


Figure 4.8. Conceptual Illustration of Error Types

From the above discussion, it may be concluded that error probability is directly related to the variability of the status variable, the amount of time covered by the observation and the setting of the mapping threshold. It is also important to make an additional observation. It is difficult to discriminate acceptable from unacceptable operation when the trend component of the status variable is equal to the mapping threshold (see Figure 4.8). Under these conditions, an IBV which is marginal, is as likely to be identified as acceptable as it is unacceptable. If it is necessary that the IBV be declared unacceptable the instant it meets the requirements of the decision rule, it may be necessary to lower the mapping threshold. This action, however, will certainly increase the probability of a Type I error.

With the general information above, it now is profitable to consider the specifics of the analysis. The material is divided into real time verification approaches and deferred time verification approaches. The reader desiring additional background may wish to consult References [7], [8] and [9].

4.8.1 Real Time Verification

Real time verification is continuously judging the status of an IBV. It might be expected that, for a given notification interval, the probability of a Type II error would be less than that of deferred time verification. This is indeed the case. However, the probability of a Type I error is greater under most circumstances.

Real time verification errors divide naturally into two areas depending on whether the status variable is analog or digital. Unfortunately, the analog case becomes quite involved and requires a great deal of knowledge about the distribution of the status variable – both from a voltage standpoint and a time standpoint. Such information is seldom available. Reasonable approximations can be made about the voltage distribution using Chebyshev's inequality if the mean and variance are known, but this provides little information about the time behavior. Further treatment of the analog case will not be

pursued except to note that *if the measured rms value of the status variable is maintained below 1/10 of the difference between the threshold value and the normal operating point of the status variable, the probability of error should be quite small.* Control of the rms value is realized in the smoothing operation.

The digital case is amenable to an analytic approach if burst errors affecting several consecutive words are negligible. Section 4.4.3.2 identified two performance criteria for digital IBV's and provided methods for their analysis and implementation. The two methods were both based on the use of parity to identify errors and employed the following two decision criteria.

- Method 1 – indicate IBV unacceptable if n or more consecutive word errors are made.
- Method 2 – indicate IBV unacceptable if n or more errors are observed during period T.

These two approaches were analyzed in Section 4.4.3.2 for the probability of error at a single decision point. How will consecutive decisions affect the probability of error since Type I and II errors are defined over time intervals? Method 2 can be analyzed by the procedure described in Section 4.8.2. This section will present a procedure for analyzing Method 1.

Both Type I and Type II error probabilities are analyzed. Beginning with Type I errors, it is desired to find the probability of identifying an acceptable IBV as unacceptable over the mission time using the criterion of n or more consecutive word errors constitutes IBV failures. Let,

p = probability of a word error (one or more bits in error)

T_m = mission time

M_m = total number of words observed during the mission

r = word rate

then,

$$M_m = \frac{T_m}{r}$$

and the probability of observing at least n consecutive words in error out of M_m observations is given by

$$P\left(n, M_m\right) = P\left(n, M_m - 1\right) + p^n(1-p)\left[1 - P\left(n, M_m - 1\right)\right] \qquad (4.8.1\text{-}1)$$

and,

$$P(\text{Type I}) = P\left(n, M_m\right) \quad .$$

Equation (4.8.1–1) can be determined only by successive values of $P\left(n,M_m\text{-}2\right)$, $P\left(n,M_m\text{-}3\right)$ etc. The recursion relation is

$$P(n, n+j) = P(n, n+j-1) + p^n\ (1-p)\left[1 - P(n, n+j-1)\right] \ ,$$
$$j = 1,2,3 \ldots . \left(M_m\text{-}n\right)$$

and

$$P(n, n) = p^n \quad .$$

Carrying out the iteration process is time consuming. If P is 1×10^{-4} or smaller, a good approximation to Equation (4.8.1–1) is

$$P\left(n,M_m\right) \simeq \left(M_m\text{-}n+1\right) p^n \quad . \qquad (4.8.1\text{–}2)$$

For example, let

$$p = 1 \times 10^{-4}$$

$$n = 10$$

$$M_m = 1{,}000$$

The approximation yields 9.91×10^{-38} and the exact solution is 9.909×10^{-38}.

Table 4.8.1–1 gives the probability of Type I errors for various values of n, M_m, and p. The entries were calculated using Equation (4.8.1–2).

An example may serve to reinforce the significance of this approach. Suppose a spacecraft is faced with a 250-day mission (perhaps a Mars fly-by) and the probability of a Type I error on a digital IBV verification is to be less than 1×10^{-10}. What value should be selected for n (the number of consecutive word errors)?

Assume:

word rate is 1×10^6 words/sec.

word error rate (p) is 1×10^{-4}.

Table 4.8.1–1. Probability of Type I Error

n \ M_m	10^6	10^7	10^8	10^9	p
5	10^{-9}	10^{-8}	10^{-7}	10^{-6}	
10	10^{-24}	10^{-23}	10^{-22}	10^{-21}	
15	10^{-39}	10^{-38}	10^{-37}	10^{-36}	10^{-3}
20	10^{-54}	10^{-53}	10^{-52}	10^{-51}	
5	10^{-14}	10^{-13}	10^{-12}	10^{-11}	
10	10^{-34}	10^{-33}	10^{-32}	10^{-31}	
15	10^{-54}	10^{-53}	10^{-52}	10^{-51}	10^{-4}
20	10^{-74}	10^{-73}	10^{-72}	10^{-71}	
5	10^{-19}	10^{-18}	10^{-17}	10^{-16}	
10	10^{-44}	10^{-43}	10^{-42}	10^{-41}	
15	10^{-69}	10^{-68}	10^{-67}	10^{-66}	10^{-5}
20	0	0	0	0	

$$\Pr(\text{Type I error}) = P(n, M_m)$$

The total number of words is

$$M_m = (250 \text{ days})(24 \text{ hrs/day})(3600 \text{ sec./hr.})(10^6 \text{ words/sec.})$$

$$M_m = 2.16 \times 10^{13}$$

From Equation (4.8.1–2),

$$P(\text{Type I}) = \left(2.16 \times 10^{13} - n + 1\right)\left(10^{-4}\right)^n$$

Now n will be very much less than M_m and it is desired to select an n such that

$$1 \times 10^{-10} > \left(2.16 \times 10^{13}\right)\left(10^{-4}\right)^n$$

or, $4.629 \times 10^{-24} > 10^{-4n}$

This will be true for n = 7. Therefore, by requiring seven consecutive word errors, the probability of a Type I error over a 250-day mission is less than 1×10^{-10}.

With all the power of this decision rule, it still has a potentially serious drawback. The probability of Type II errors is extremely large unless the word error rate becomes significantly high. The method could not, then, be used to distinguish drifts in error rate as large as an order of magnitude. If interest lies in detecting "hard" failures where circuits lock up rather then determining error rate, then the method will be satisfactory for Type II error detection as well. When a "hard" failure occurs, it is possible that all words will be in error and that this will be detected in the minimum time, n. However, error detection schemes are not this good and a finite probability exists that a word will be identified as being error free. With this in mind, define

τ = notification time

M = total number of words observed in τ

and let all other variables be the same as those for Type I errors. Then,

$$M = \frac{\tau}{r}$$

and from Equation (4.8.1–1),

P_r(Type II error) = 1 – P(n, M)

with M_m replaced by M. The approximation cannot be used for the low values of word error rate and the equation must be directly applied. Some representative values of probability of Type II error for M = 1000 are shown in Table 4.8.1–2. Note that as n increases (for a fixed value of p), Pr(Type II) dramatically increases until it approaches about 0.1. Also, as p increases beyond 0.6, Pr(Type II) rapidly decreases.

4.8.2 Deferred Time Verification

Deferred time verification examines an IBV on a periodic basis and makes status decisions based on this brief sample. It will be assumed that error probabilities at each verification instance are independent and constant over time. Define,

p_m = Pr(miss error) = probability that, at a single verification instance, an unacceptable IBV is identified as acceptable

p_f = Pr(false alarm) = probability that, at a single verification instance, an acceptable IBV is identified as unacceptable.

With these introductory remarks, the remainder of the section will address the specifics for Type I and II error probabilities.

Table 4.8.1–2. Probability of Type II Error

M = 1000

n \ p	0.5	0.6	0.7	0.8	0.9
5	1.6×10^{-7}	10^{-10}			
10	6.2×10^{-2}	10^{-1}	2.2×10^{-4}	5×10^{-10}	
15	9.85×10^{-1}	8.3×10^{-1}	2.5×10^{-1}	9.2×10^{-4}	1.1×10^{-9}
20		9.85×10^{-1}	7.9×10^{-1}	1.03×10^{-2}	5.5×10^{-6}
25			9.6×10^{-1}	4.8×10^{-1}	8.3×10^{-4}
30			9.94×10^{-1}	7.9×10^{-1}	1.6×10^{-2}
35				9.25×10^{-1}	8.7×10^{-2}
40				9.75×10^{-2}	2.4×10^{-1}

Pr(Type II error) = 1 - P(n, M)

4.8.2.1 Type I Error Probability

This probability is quite straightforward. Let

T_m = mission time

a = time between verification

m = number of verification instances in T_m

Then

$$m = \frac{T_m}{a} \qquad (4.8.2-1)$$

Since m must be an integer, it may be necessary to work with the two values that span T_m/a if it is not integer.

The probability of a Type I error is then

$$P_I = 1-(1-p_f)^m$$

Figure 4.8.2.1 is a graph of p_f vs. P_I for several values of m. As an example, let p_f be 0.001 and require that P_I must be less than 0.1. What is the maximum value of m that will satisfy this requirement? From the figure, m = 100 corresponds to P_I of 0.11 and m = 50 to P_I = 0.055. Interpolating between the values of m, a value of 90 for m should meet the requirement. The corresponding value for a can be determined from Equation (4.8.2–1).

4.8.2.2 Type II Error Probability

The Type II error probability is somewhat more involved than that of Type I. This is due to the fact that the time between verification intervals (a) can be less than or greater than the notification time, τ. If a is less than τ the problem formulation is the same as that of the Type I error with variables changed. Let

τ = notification time

N = number of verification instances.

Then

$$N = \frac{\tau}{a}$$

where a is still the time between verifications. The probability of a Type II error is then

$$P_{II} = p_m^N, a<\tau$$

A plot of this equation is shown in Figure 4.8.2.2–1 for values of p_m and N. As an example, let

p_m = 0.3

τ = 10

and require that P_{II} must be less than 0.01. From the figure, N must be at least four. The value of a which corresponds to this quantity is 2.5.

There is a distinct possibility that the time between verifications will be greater than the notification time. For instance, it would not be unusual for the time between verification to be 12 hours and the failure notification time to be three minutes. In fact, this would be a typical application of deferred time verification. For most requirements on notification time, values of a less than τ comes close to real time verification. For example, under the digital decision criteria described in Section 4.8, method 2 would be solved using the approach of a less than τ.

If a is greater than τ, there are two possible events which can occur following a failure of the IBV:

- a single verification instance occurs within τ, or
- no verification occurs within τ.

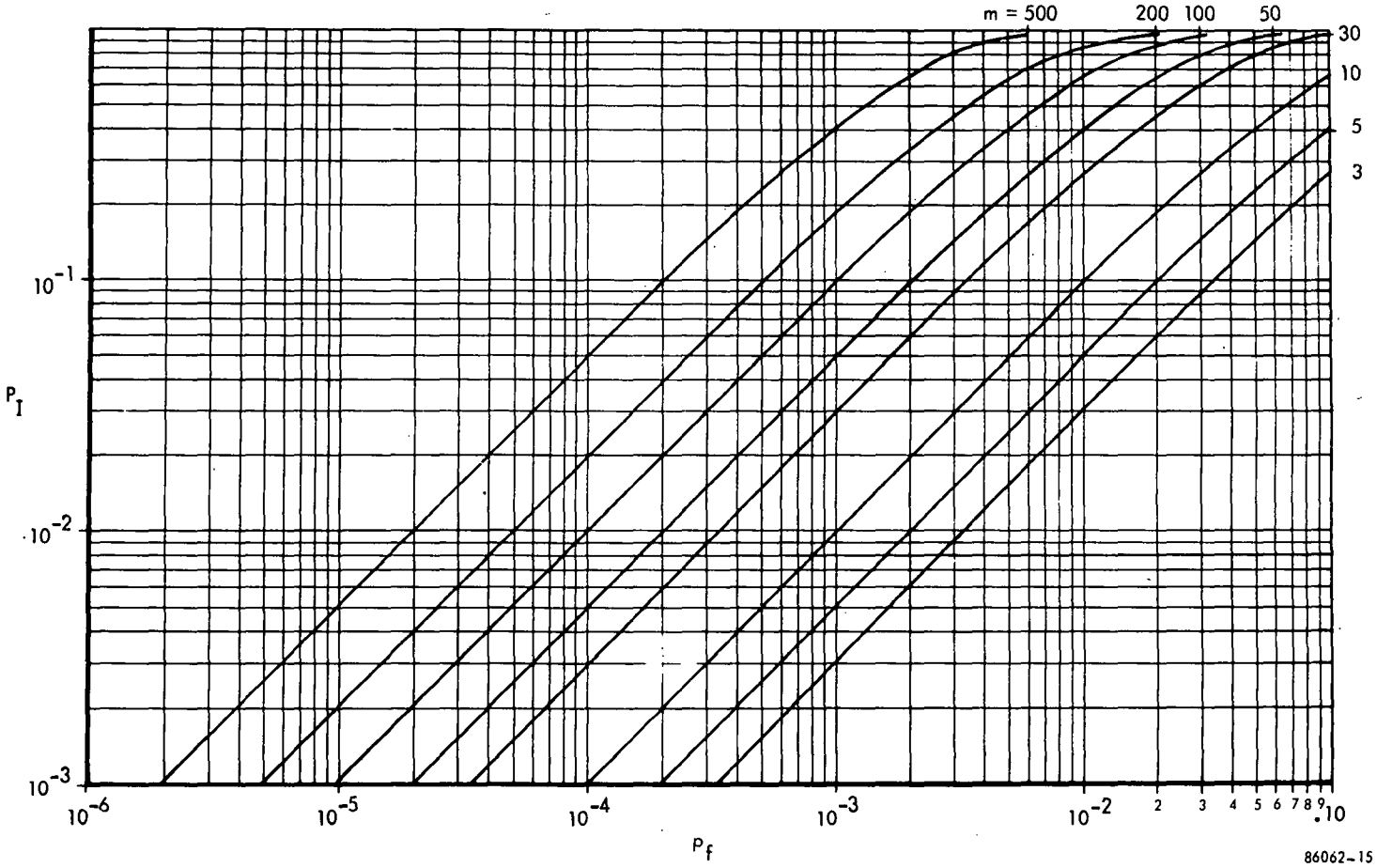


Figure 4.8.2.1. Probability of Type I Error vs. Probability of False Alarm

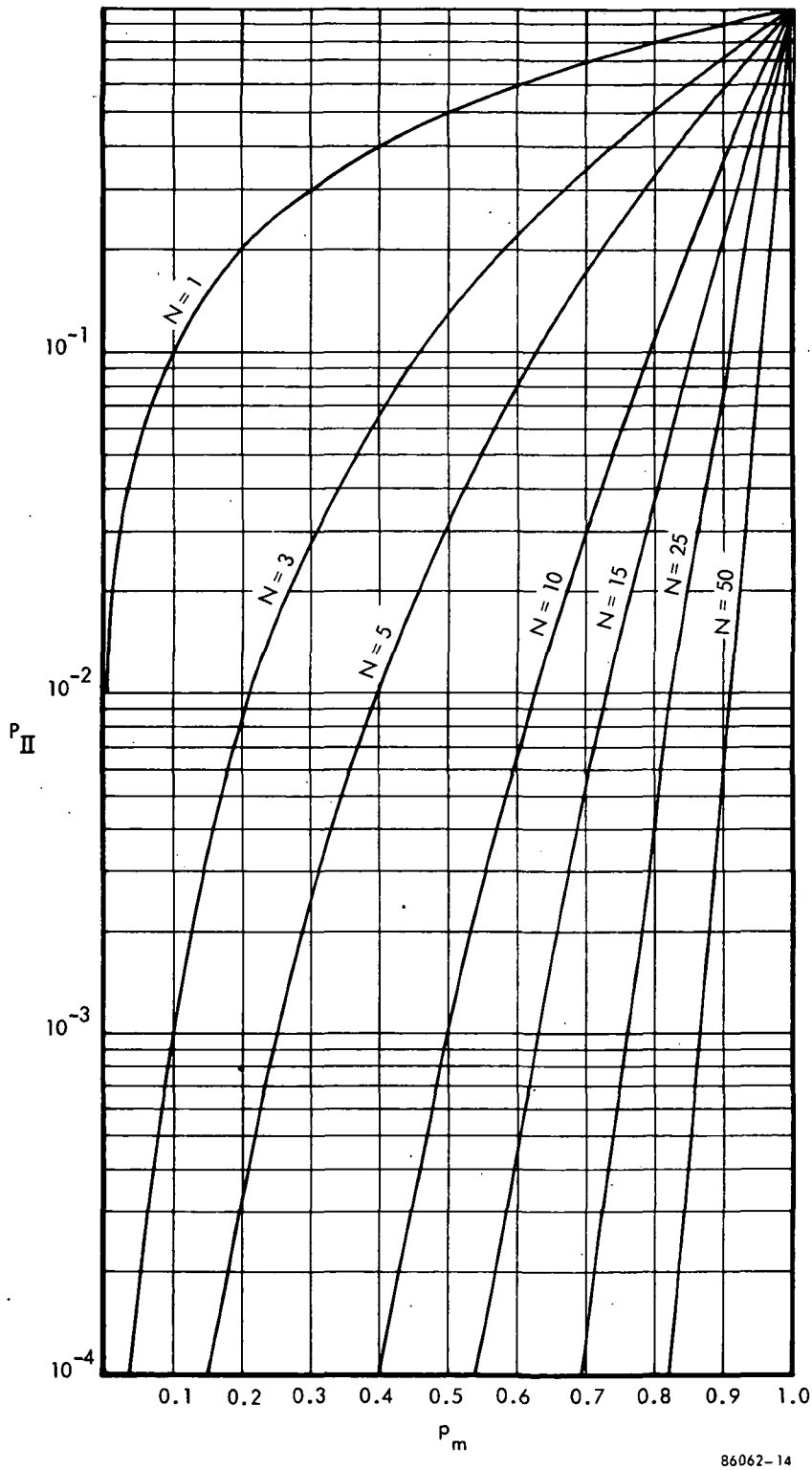


Figure 4.8.2.2–1. Probability of Committing a Type II Error When $\tau > a$

It is impossible for two or more verification events to occur. Let g be the number of verification instances which occur during τ. The probability of a Type II error is

$$P_{II} = 1 - \left[\left(1 - p_m\right) \Pr(g = 1) + 0 \Pr(g = 0)\right]$$

$$P_{II} = 1 - \left(1 - p_m\right) \Pr(g = 1) \quad . \qquad (4.8.2.2\text{–}1)$$

The probability distribution on g depends on the reliability of the IBV. This is as it should be since values of a greater than τ admit to a risk of the IBV failing and that event not being indicated within the notification interval. Assuming the exponential failure distribution for the IBV and conditioning the results on the given information that the IBV has failed in interval a,

$$\Pr(g = 1) = 1 - \frac{1 - \exp\left[-(a-\tau)/\theta\right]}{1 - \exp\left[-a/\theta\right]} \quad , a > \tau \qquad (4.8.2.2\text{–}2)$$

where θ is the MTBF of the IBV. Note that as a approaches τ, Pr(g = 1) approaches unity.

Figure 4.8.2.2–2 is a plot of Equation (4.8.2.2–2) normalized to a/θ and τ/a. As an example, if θ= 100 hrs., τ= 1/2 hr. and a = 1.5 hrs. Then a/θ = 0.015 and τ/a = 0.333. Interpolating between τ/a = 0.3 and τ/a = 0.4, the value of Pr(g = 1) is approximately 0.33. From Equation (4.8.2.2–1), the probability of Type II error would then be calculated as

$$P_{II} = 1 - \left(1 - p_m\right)\left(0.33\right) \quad .$$

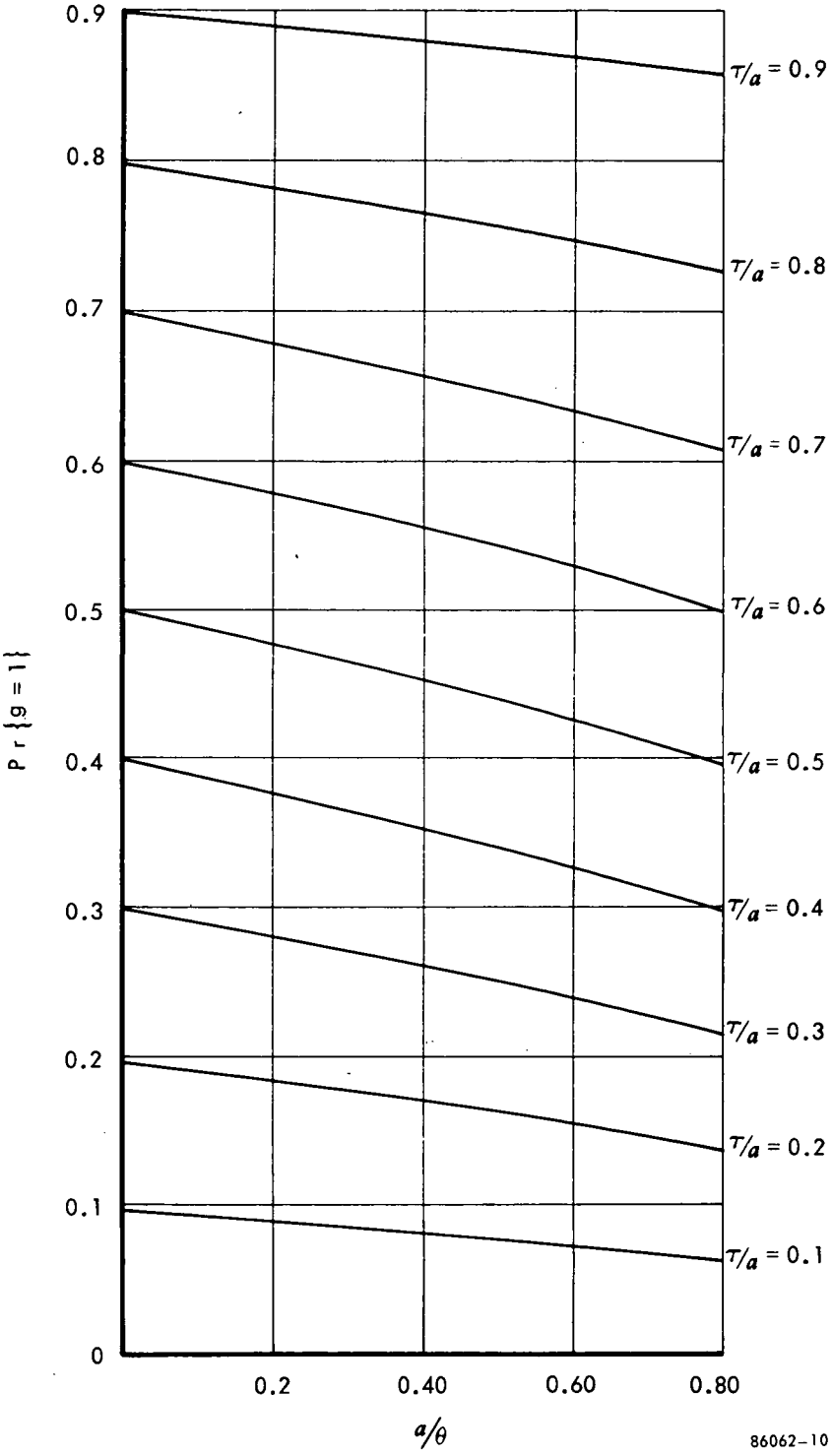


Figure 4.8.2.2–2. Probability of Getting a Chance to Observe a Failed IBV When $\tau < a$

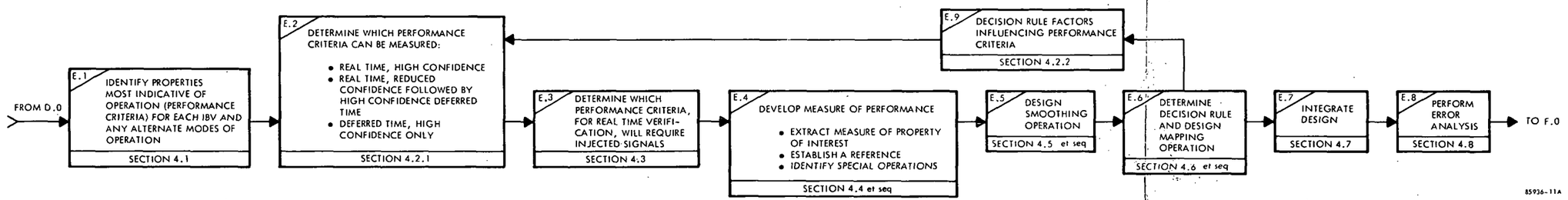
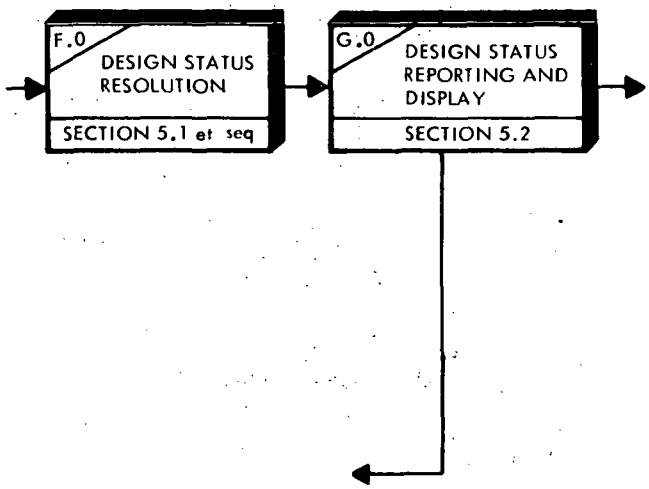


Figure 4.0–3. Status Development Design Process

SECTION 5.0

STATUS HANDLING



5.0 STATUS HANDLING

Recapping, Section 3.0 addressed the general system level problem to identify specific requirements for each Item Being Verified (IBV). Section 4.0 then described the verification design of each IBV. What remains is the process of treating status indications from many IBV's in a system and the reporting of these results. As such, this section will return to the general system problem of synthesizing the individual status indications.

5.1 STATUS RESOLUTION

BLOCK F.0

This section describes the efforts to be accomplished in Block F.0 of Figure 1.4–1 (Page 14), viz., the concepts and design approaches to status resolution.

The requirement for status resolution stems from the premise underlying the entire approach to redundancy verification. *This premise is that the status of an IBV is the same as the status of its output, i.e., if the output is satisfactory, the IBV must be performing acceptably.* The premise tacitly assumes that the IBV is receiving an acceptable input. What if it is not? The chances are slim that the IBV output will be acceptable, even if the IBV itself is performing perfectly. Then, *status indications from status development are in reality conditional indications, for they cannot be identified as IBV status until the status of the input is known.* That is, absolute status statements can only be made pending investigation of IBV input *status* (or as a minimum, input *admissibility*). This seems trivial enough until consideration is given to a failure in the middle of a series chain of IBV's. Typically, the conditional status of all IBV's to the right of the failed IBV (in a functional flow sense) will also indicate unacceptable. From these indications, the IBV which is truly at fault must be identified. To accomplish this on an automated basis would require a device that "knew" the information flow and could deduce that, with high probability, the first IBV in the string of "unacceptable" IBV's is the item at fault. Conditional status of the remaining IBV's would then be declared unconditionally acceptable. This is the function of status resolution and it must be performed whenever conditional status indications cannot be immediately interpreted as IBV status. Figure 5.1 illustrates the conceptual operation of status resolution. Note that the status resolution processor remains idle until a status change is indicated at the OR gate. This feature would allow status resolution to be time shared.

There are two situations where status resolution may not be required. They are:

a. when input/output and output/output reference methods (see Section 4.4.2) are used at each IBV, or

b. when an injected signal is used at the input to each IBV.

So long as the IBV input remains admissible, reference methods which depend on the value of the input, or compare the outputs of two identical IBV's, are usually self-resolving. That is, these reference methods will not allow an IBV status to become unacceptable simply because input signal status is unacceptable (but still admissible). Whether these reference methods will actually eliminate the need for status resolution should be carefully checked in each application. It will be necessary to determine if dominant failure modes of the input signal will result in a nonadmissible signal. Also, it is not prudent to build up

exceptionally long chains of IBV's (using these reference methods) without any reliance on resolution.

It must be recognized that there is an underlying assumption in the above discussion, namely the independence of element status. While it has been admitted that conditional status indications can ripple down an on-line chain, it has also been implied that, so long as the conditional status of an IBV is "good," that IBV output cannot, with high probability, degrade to a state which will cause the succeeding IBV to enter an unacceptable state. Stated another way, if the conditional status of an IBV indicates acceptable operation and the conditional status of the immediately succeeding IBV indicates unacceptable operation, the fault must be in the second IBV. The problem usually comes about when two communicating IBV's drift to opposite sides of the operating tolerance band.

There is a natural tendency to compensate for the risk of violating status independence by enlarging the range of the status variable in the region of unacceptable operation. However, this approach will usually increase the probability of identifying an IBV as unacceptable when it is actually acceptable.

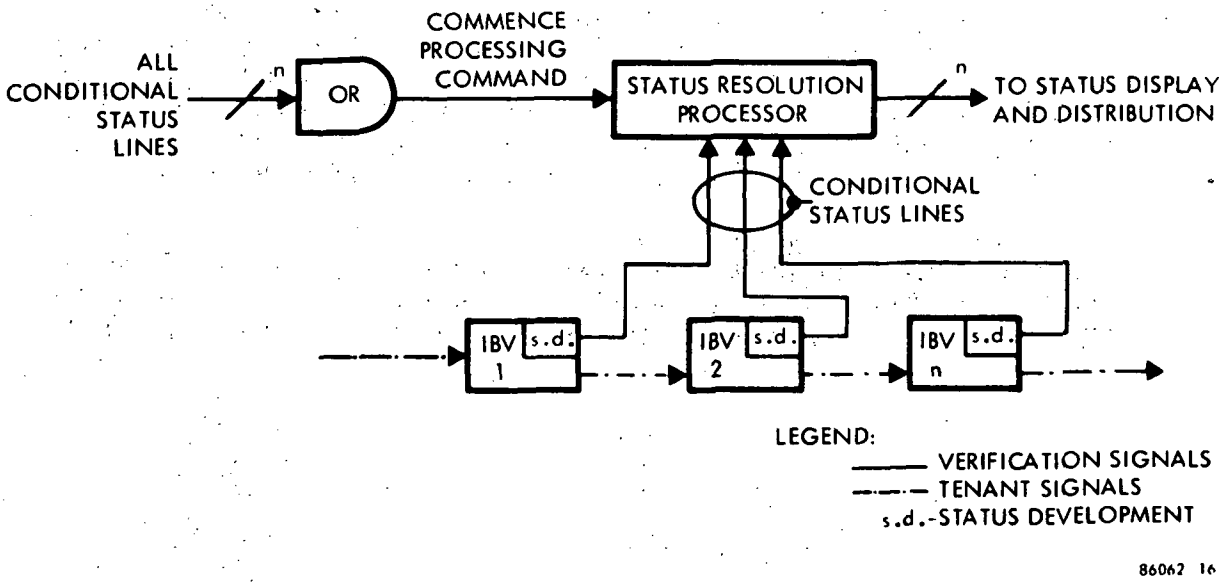


Figure 5.1. Conceptual Operation of Status Resolution

Status independence is affected by the number of status levels. As the quantity of status levels increases, the range of the status variable will be divided into additional mapping regions. This results in smaller regions which are more susceptible to operational drift. In addition, it will be quite difficult to effect resolution if an IBV status is "marginal". Consequently, *the problem of preserving status independence becomes quite crucial if more than two levels of status are required.* One way around this problem is to display multilevel status and ask the automated system to act on two levels which are defined by partitioning the multilevels.

If each IBV in the principal system employed a simulative signal at its input, status resolution would not be necessary since a known input is being supplied. Status indications can then only result from IBV's whose input status is known. Extrapolating this result, it can then be stated that *status resolution need only be applied between signal injection points in a series chain of IBV's.* As a result, resolution patterns can be developed from the partitioned principal system based on signal injection points. It should also be noted that *status resolution is not capable of indicating the truly failed IBV in a closed loop.* If it is required to identify elements within a loop, injected signals must be used.

From the functional description of status resolution, the obvious method of implementation is a digital processor.* Less costly hardware approaches are feasible, however, for systems which do not change operational configurations. The hardware approach has the advantage of simplicity and reliability. The processor approach is flexible and, if a processor is required for other functions, provides compatability. The final choice could be a combination of hardware and software.

5.2 DESIGNING STATUS REPORTING AND DISPLAY

With status resolution incorporated, firm status statements can be made for each IBV. These statements must now be incorporated with group deduction results to achieve the complement of principal system status indications.

At this stage of design, the status indications are a collection of voltage levels or encoded digital words. It now remains to convert these indications into status displays and/or signals capable of driving automatic reconfiguration equipment (when required). This operation will involve level converters, decoders and drivers to achieve compatability. It is entirely possible to deluge a console operator with status information. Logic operations may then be included to provide status summary indications for selected sections of the principal system.

Displaying of status can be accomplished with status boards, simple incandescent lamps or CRT displays. The choice is almost exclusively one of operations and cost. It should be remembered, however, that some approaches are intended to provide information on which to base decisions and others to indicate a specific action that an individual is to take. If the latter is the case, the action to be taken should be clearly indicated and there should be assurances that the individual will be able, upon notification, to initiate the act. If he is expected to be kept busy at other tasks, he may not be able to respond within the allotted time. Indications involving safety or life support should usually not require response of an individual since there exists a possibility he may be incapacitated and cannot respond.

---

*It should be noted that the entire concept of status resolution is based on the assumption of single failure occurrences, i.e., only one IBV will become unacceptable during the short period of examination. If this assumption is violated, ambiguities will likely result.

SECTION 6.0

ADDITIONAL CONSIDERATIONS

6.0 ADDITIONAL CONSIDERATIONS

This section is intended to discuss considerations which are apart from, but far from incidental to, the design process. The first of these considerations is the cost of automated redundancy verification. The second consideration deals with design approaches to typical element isolating devices such as switches, relays and diodes.

6.1 COST OF VERIFICATION

This section provides a cost estimate for realizing automated redundancy verification. The estimate is intended as a guideline for planning purposes during the conceptual stages of a system. As a system design evolves and more detailed information is available, cost estimates should obviously be refined.

The following estimates represent the total cost of including automated redundancy verification in terms of the design, development and hardware cost of the base principal system.

Ground Station	15% to 30%	of principal system base cost
Space Station/Vehicle	22% to 60%	

Principal system *base costs* are associated with verified equipment only and exclude costs of all items which do not contribute directly to their functional operation. Therefore, costs of spacecraft airframes and the like would not be included under the base principal system cost. Also excluded from the base cost would be costs for documentation as well as costs associated with design and human engineering of principal system consoles and displays.

The cost estimate assume:

- real time verification, and
- verification is part of the initial principal system planning and design.

As a class, command and control systems will tend toward the upper portion of the estimates. Digital implementations using parity should tend toward the lower half of the estimates.

For purposes of comparison, deferred time verification using simulative signals should always fall on the low extreme of the estimate.

6.2 SWITCHES, RELAYS AND DIODES

Switches, relays and diodes are frequently used to achieve failure independence of redundant elements. As such, they represent an intervening device in the functional flow and can affect the status of a redundant network. Verification of these devices is, then, a natural concern.

When used in a design, switches, relays and diodes (isolation devices) are, without exception, operating in the conducting or "shorted" mode as their normal state. Difficulty

arises in attempting to determine if the device has failed short, for if this occurs, it will not be able to perform a state change when required.

It has been demonstrated in Reference [2] that the only method of assuring that such devices have not failed "short" is to exercise them periodically, i.e., force them to change state. This usually requires interruption of the principal system mission but will verify device operation. There is, unfortunately no way to verify the devices in real time.

As with any deferred time verification, observing proper operation at one instance of verification cannot assure proper operation in the future. Therefore, the practical approach to assuring operation of isolation device is to increase ones confidence that they are operating properly, i.e., improve their reliability. This can be achieved by either using hi-reliability parts or employing redundancy. In either event, Reference [1] may be consulted as an aid to the design and selection of an alternative.

SECTION 7.0

EXAMPLES

7.0 EXAMPLES

This section contains four examples of redundancy selected from equipment currently in use. The examples have been selected to offer a wide variety of situations to which redundancy verification can be applied. In addition, each example emphasizes a particular area of the redundancy verification problem.

Analysis and critique contained in the examples are intended to illustrate the design process described in the handbook. Example 1 contains a step-by-step account of the design process with the remaining examples being treated in summary form.

Since the examples represent single instances of redundancy, i.e., individual redundancy networks, the processes of partitioning and status resolution are not described. Presumably, partitioning was initially performed on the principal system to identify the networks. Whether it is a sound practice to verify to the level indicated by the example networks can only be determined in the context of adjacent networks, their relationships to the example network and the overall function to be performed. In any event, this decision is part of the overall system problem and cannot be addressed here. Recommendations for or against the use of the example redundancy approaches will depend on the relative weights of mission requirements assigned by the designer. The examples will, however, contain critiques of the redundancy design features relative to verification.

7.1 EXAMPLE 1 – SINGLE STAGE QUADDED VALVE NETWORK

This example develops a verification approach to quadded fuel valves. The valve network was selected as an example to illustrate group and failure mode approaches. In addition, the example identifies the verification methodology to mechanical disciplines.

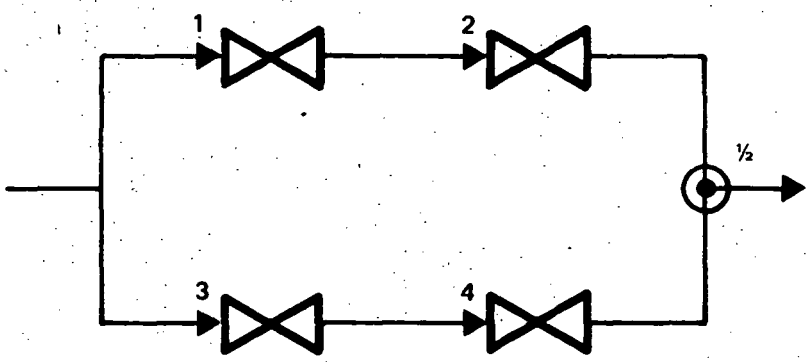
7.1.1 Network Description

The quadded valve network shown in Figure 7.1.1(a) is used for fuel control aboard a space vehicle. The quad configuration has been selected to minimize effects of valve failure modes within the network. The network is the only means of fuel control between tankage and motor.

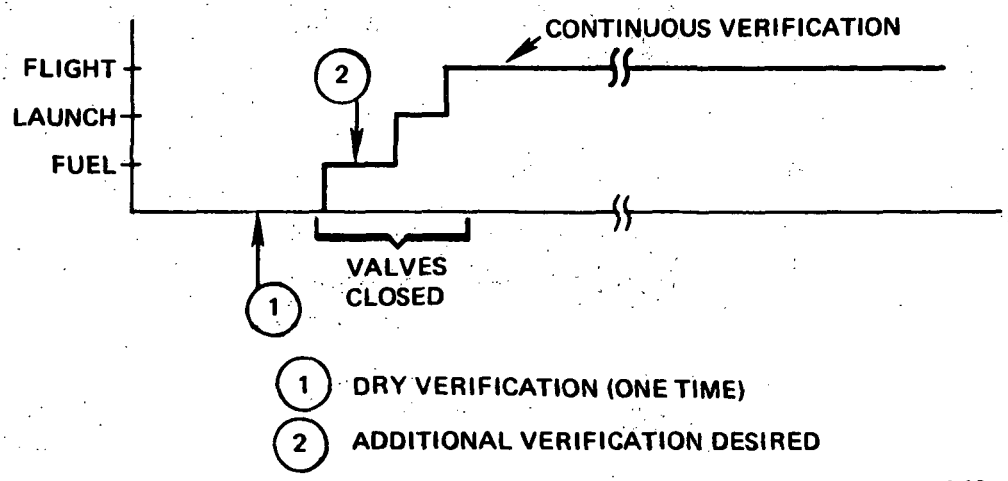
The network redundancy will be verified once before fueling and verification is to be continuous after launch. It would be desirable to perform an additional verification after fueling and before launch. The mission profile is shown in Figure 7.1.1(b).

The valves are used to control a maneuvering motor and will thus be opened and closed many times during the course of flight. All valves are identical and driven by the same command device. Each has but two positions – open and closed – and can fail (lock up) in either of these positions.

The valves are critical and it is desired to know the status of each. Since the valves are identical, this is equivalent to knowing how many valves are operative in each of the redundant paths. A valve is nonoperative if it remains closed (or subsequently moves closed) while an "open" command is being supplied. Conversely, a valve is also nonoperative if it remains open (or subsequently moves open) while a "close" command is being supplied.



(a) NETWORK FLOW DIAGRAM



(b) NETWORK MISSION PROFILE

Figure 7.1.1. Example 1 – Description Diagram

7.1.2 Initial Assessment

The first consideration should be the importance of the valve failure modes. From fueling until the end of launch, the valve network must remain closed (see Figure 7.1.1(b). During this period, the only failure mode of interest is a valve opening: a valve locking in a closed position can certainly cause no problems. If this were the only portion of the mission, closed failure modes could be ignored and the problem would be much simpler. Once flight begins, however, the valves will be opened and closed virtually at random. The network cannot fail in either mode since both failure modes are safety hazards. Both failure modes must then be considered and they must be assumed equally important.

With both failure modes being equally important, the next consideration is their influence on verification. This is best understood with the aid of an additional illustration. Figure 7.1.2 is an equivalent network diagram redrawn from Figure 7.1.1(a).* The notation is as follows:

C_1 is the event that valve 1 fails closed,
C_2, valve 2 fails closed, etc.
O_1 is the event that valve 1 fails open,
O_2, valve 2 fails open, etc.

The diagram may be interpreted as the occurrence of an event in one of the blocks prevents passage through that block. So long as a path can be found through the network from point A to point B, the network is performing satisfactorily. Note that if either valve 1 or valve 2 fails closed, the top path can no longer be traversed, i.e., one redundancy path is inoperative. On the other hand, if either (but not both) valve 1 or valve 2 fails open, the redundancy path is still operative. This finding is significant. It means that if the crew desires a true indication of available redundancy during flight, they must not simply be informed that a valve is inoperative but *also apprised of its failure mode*. Knowing that valve 2 failed could mean one of *two* things.

a. It failed closed and half the redundancy is lost.

b. It failed open, still leaving both redundant paths operative.

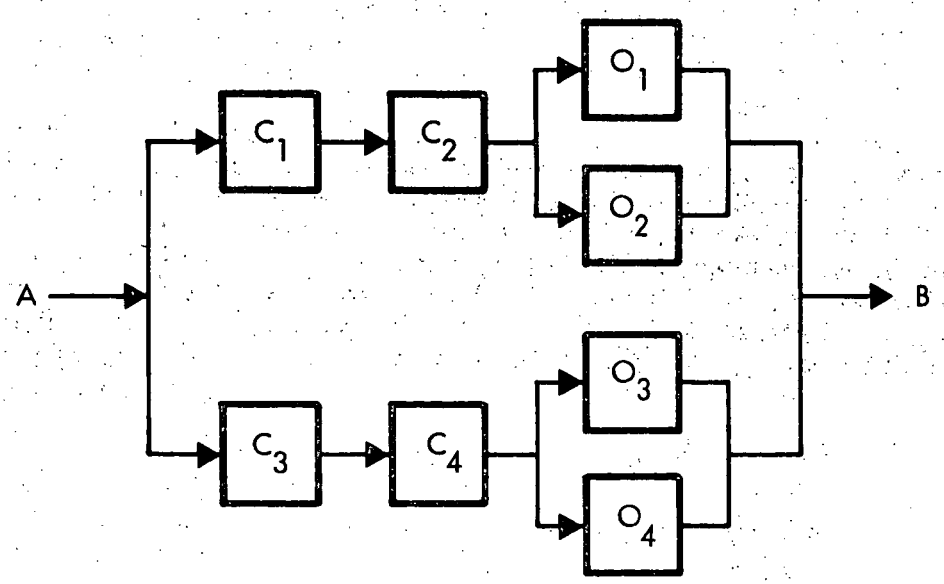


Figure 7.1.2. Example 1 – Equivalent Network Diagram

*The illustration is not rigorously correct, for once a valve fails closed, it cannot fail open. Properly interpreted, the diagram is a good mental aid.

The value of the equivalent network diagram should by now be obvious. It is often beneficial to view redundancy verification problems with the aid of such a diagram. For example, each redundant path in the network consists of three simple sets: two having redundancy degree zero and one having redundancy degree one.

It should be noted that the *extension of status information into failure modes is an exception rather than the rule*. This example was selected specifically to illustrate the possibility of its occurrence. There are two reasons why it occurs here. First, the valves are binary devices. Second, they can fail in such a way as to allow a second valve in series to continue performing the function of both. That is, if valve 2 fails open, it appears as just another length of fuel line. Valve 1 can then continue operating as if valve 2 never existed. (This can be seen in the equivalent diagram.) More complex pieces of equipment seldom exhibit this second feature and status indications need only be concerned with whether an IBV is not operating: not with the manner in which it ceased to operate.

7.1.3 Verification Analysis

Analysis of the network will follow the design process discussed in Section 3.0 and illustrated in Figure 3.2, page 51. With available information, the analysis will then extend to the first four blocks of Figure 4.0–3, page 111.

Establish Level of Verification BLOCK D.1

Since the single redundancy network is being addressed, determination of level of verification is not as significant for this example as it would be when an entire principal system was being considered. The network should, however, be identified as a group or set at this stage since subsequent design alternatives depend on this identification.

The single network will either be a stand-alone set or a first level group; which implies that there must be at least one point of verification involved. If the network is to be classed as a set, the pair of valves in each branch must, together, be considered an element. (This is the only arrangement which will result in common input and common output nodes.) Referring back to the initial assessment in Section 7.1.2, a little thought will show that staus information supplied on elements thus defined will provide a degree of redundancy verification. This approach cannot, however, supply the detailed status information desired in Section 7.1.1 above. Each valve must then be considered an element and the network is then obviously a first level group.

Establish the Use of Real Time or Deferred Time Verification BLOCK D.2

From the network description in Section 7.1.1, both real time and deferred time verification are required. The necessity of real time verification results from the requirement of continuous verification during flight. During this period of time, the mission cannot be interrupted to effect verification on the valves. Deferred time verification is necessary due to the single dry verification prior to fueling and, if it can be achieved, the one-time verification between fueling and launch.

Since both forms of verification are required, the final design should incorporate an approach which is applicable in either case. This will avoid two separate verification schemes.

Determine Time Profiles for Deferred Time Circumstances BLOCK D.3

The network time profile is implied by the mission profile given in Figure 7.1.1(b). From this profile, it can be seen that, during the single dry verification, the network can be exercised and simulative signals can be injected without interrupting the mission. This is the best of possible circumstances.

The additional deferred time verification point after fueling is quite another problem. The network cannot be exercised since the valves must remain closed. This requirement also rules out the use of simulative signals.

Indicate Confidence Levels BLOCK D.4

Since both deferred time and real time verifications are required, there will likely be a confidence level assigned to each. In addition, the added deferred time verification point after fueling cannot employ exercising of the valves. It is not likely then, that this verification point can achieve the same confidence as that of the dry verification. Actually, the two points will determine different performance features. The dry verification would be well suited for determining valve dynamics. The "wet" verification could be used to check for leakage.

Since the valves have but two operating states, it should not be difficult, at least in theory, to extract sufficient information to represent operational integrity. The limiting factors will be the availability of transducers and the physical restrictions and limitations of network hardware.

It will next be necessary to establish quantitative error probability requirements. These requirements are typically budgeted from overall system requirements and depend on notification time and mission time. As such, they cannot be determined for this example. The consequences of the errors, however, *can* be stated and form an important input to quantitative error determination.

During the period before launch (deferred time verification), Type I errors will cause unnecessary holds. This is costly, but can often be considered an inconvenience unless it results in missing a launch window. Type II errors, on the other hand, will result in the launch of a vehicle which does not contain a full complement of redundancy. This is obviously quite serious and a potential safety hazard.

During flight, Type I errors can cause unnecessary aborts. This is extremely costly in terms of lost man-hours, equipment and incompleted missions. Depending on the circumstances, Type II errors may also be quite serious. If the crew depends on redundancy status results before committing to a critical maneuver, a Type II error may lead them to believe it is safe to continue. Whereas, if they had known that redundancy was degraded, they would never have executed the maneuver. This is a distinct safety hazard.

Suppose, however, that the valve network fails in the middle of a burn. The crew is already committed and the fact that redundancy verification equipment continues to indicate redundancy is still present is somewhat academic. The crew can quickly tell that it is not and there will probably be little that they can do to remedy the situation.

Indicate Signal Injection Required for Deferred Time Verification BLOCK D.5

The prefueling deferred time verification will require an exercising signal to determine valve dynamics and achieve high confidence in status results. Depending on the countdown event sequence prior to verification, this signal could actually be the tenant signal or it may be an injected signal to simulate that signal. Whether an injected signal is inserted directly into the network or several functions earlier, depends again, on the event history and the preceding functions. In any case, determination of signal injection points is part of a larger system problem. If a simulative signal is injected at the network input, status resolution will not be required.

During the second deferred time verification, the network cannot be exercised and will not be able to use a simulative signal. In fact, precautions should be taken to "lock out" all simulative signals from this point onward.

In summary, the first deferred time verification will require a dynamic simulative signal. The second deferred time verification cannot use a simulative signal.

Determine Impact of Other IBV Properties BLOCK D.6

IBV's have as yet not been determined, but with present knowledge of the valves (network elements), conclusions can be drawn regardless of what constitutes an IBV. The network, or portions thereof, cannot be independent of its element properties.

Of concern in this portion of the design are any peculiarities of operation which can cause the status of output signals to differ from IBV status. Two major causes of this condition are multimode operation and nonlinearities.

The network, and the valves therein, have but one function: control fuel to the maneuvering motor. They always perform this function in the same manner. The valves then, have a single operating mode and multimode operation is of no concern.

With respect to nonlinearities, the results are not as simple. The valves *are* nonlinear devices. When a valve is closed, there is no way to determine if it has not failed in the closed mode until it is commanded to open. Conversely, there is no way to determine it has failed in an open mode until it is commanded to close. There exists, then, periods of time when the element (valve) is not being exercised. Consequently, simply knowing the valve is open when it should be, is no guarantee of its operational integrity.

Using an exercising simulative signal in the first deferred time verification will allow both valve states to be verified. Until some time during flight when the maneuvering motor is first burned, this is the last time the closed failure mode can be verified. (The second deferred time verification can only verify that the valves have opened.)

In-flight verification is impacted the most by the nonlinearity problem. If the motor is used frequently, it is reasonable to assume that verification is being performed real time since the valves are being exercised. If this is not the case, it may not be possible to achieve real time verification. Details of this problem will be addressed later under Real Time Injected Signals.

From this point, the design process divides into two branches: one for stand-alone sets and the other for groups. Since this network has been identified as a group, the latter branch will be taken.

Determine Group Deduction Schemes BLOCK D.10

The purpose of this design activity is to determine a scheme for indicating status of the four valves using, ideally, a single verification point at the network output. One of the three deduction schemes (iterative, logical, exhaustive), or a combination thereof, will be required. Once the deduction scheme is selected, IBV identification and verification points will fall out automatically.

The strategy here is to provide maximum status information with a minimum number of verification points. It is then reasonable to begin the analysis assuming a single verification point at the group output and determine what schemes will provide suitable status information. The number of verification points may then be increased and the process repeated. In this way, it is often possible to arrive at several candidate schemes, each of which may be pursued if desired. It is not intended that the analysis delve into *what* is being measured since this will be investigated in detail as part of Performance Measures. The objective is to establish a "reasonable" approach based on information at hand. It should be recognized that additional information may result in modifications.

The group under consideration is quite simple and will not require detail analysis or consideration of alternative approaches. Beginning with a single verification point at the group output, iterative deduction is usually a likely candidate under these circumstances. Iterative deduction requires the existence of sets in the group so that substitution can be effected. The network under consideration contains no sets, therefore the iterative scheme can be eliminated.

The next choice should be logical deduction since exhaustive deduction, by definition, requires the most verification points. It is obvious that logical deduction cannot provide element status using the information provided at the group output.

What if the number of verification points is increased to two? By the network symmetry (see Figure 4.1.1(a)), these two points should be located between valves 1 and 2 and between valves 3 and 4. Using results from these two points, can status of each valve be deduced? The answer to this question will require a brief discussion of fluid properties which might serve as status indications.

There are two fundamental fluid properties which should be considered. These are pressure and flow (or flow rate). If it is assumed that throttling takes place down-flow from the network (which should be entirely reasonable for this example), pressure will likely be a poor status indicator for valves which are either open or closed. For example, suppose valve 1 is closed (Figure 7.1.1(a)) and all other valves are open. If the flow rate is not great and the network output junction does not form a small angle into the common line, pressure on the right of valve 1 will be about the same as that on the left.

Returning to the two verification points, suppose flow transducers were inserted for verification. These would indicate the condition of one or both valves being closed in a branch of the network. They would not, however, indicate the condition of one valve being

locked open. From the preceding discussion, pressure transducers located at these points would result in a similar problem. In fact, a combination of flow and pressure transducers would not provide sufficient information to deduce status of all four valves.

The next step is to consider three verification points. Two of these points must be located as before with the third at the common output line. It is not difficult to see that this approach suffers from the same drawbacks as the one above.

If three verification points will not allow valve status to be deduced, logical deduction, along with iterative deduction must also be eliminated as a candidate. This leaves exhaustive deduction and a verification point must be associated with each valve. As a result, it is now possible to identify the individual valves (network elements) as IBV's. This important piece of information completes the group deduction portion of the design.

Since exhaustive deduction is being used, it will not be necessary to design group logic and control to achieve deduction. *If status resolution is required, however, it will still be necessary to provide a device which can assimilate valve status indications and provide a status indication for the entire group,* i.e., indicate whether redundancy is present or not.

Determine if Outputs are Distinguishable BLOCKS D.7, D.8, D.9

Figure 3.2, Page 51, indicates that if exhaustive deduction is used, the process branches back to the stand-alone set. While each valve (IBV) is a degenerate case of a stand-alone set, the analyses relative to distinguishable outputs, variation of output and set/element IBV identification do not apply to a single element. These analyses are intended to provide further breakdown of IBV identification for groups which contain simple sets and may, therefore, be omitted in this example.

Determine Approach to Off-Line Elements BLOCK D.12

There are no off-line elements in the example network and this analysis may be omitted.

Summarize Design Data BLOCK D.13

A summary of the design conclusions is listed below. Since a single network is being considered, the list is shorter than that which would result from a complete design.

- The network is a first level group.
- Exhaustive deduction is to be used.
- Each element (valve) in the group is an IBV.
- Both deferred and real time verification are to be employed.
- A simulative input signal will be required for the dry deferred time verification.

Identify Interfaces BLOCK D.14

With respect to the network, the following interfaces will exist.

- Signal injection point at the input. (This point could, however, exist several functions prior to the valve network.)
- Signal extraction point at each valve.

With the design data summarized and interfaces identified, design characterization is complete. The next major phase of the process is design of status development. The remaining network analysis will follow the design process discussed in Section 4.0 and illustrated in Figure 4.0–3, page 111. The analysis will concentrate on the first four blocks in the process.

Identify Properties Indicative of Operation BLOCK E.1

The best place to begin this analysis is to establish a concise functional description of the IBV. This will not only define the purpose of the IBV but also its inputs and outputs (the latter not always being as simple as it may seem). Figure 7.1.3 shows a functional diagram of a single valve in the network. As can be seen, its sole purpose is to change the fuel flow (here, *off* or *on*) at the direction of a bilevel electrical command. (Note that a fuel head is *not* the valve *functional* input.) From this simple analysis, the performance criterion can be stated.

> Allow fuel to flow at maximum rate upon receipt of an open command *and* shut off fuel flow upon receipt of a close command.

This is the only pertinent criterion for the valve.

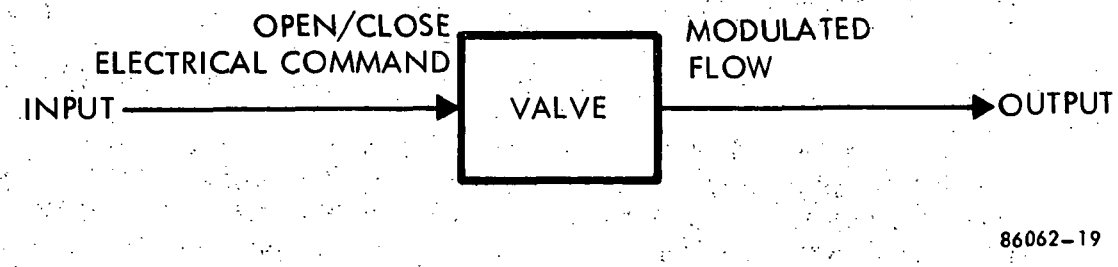


Figure 7.1.3. Single Valve Functional Diagram

Determine Which Criteria are Measured Real Time and Deferred Time BLOCK E.2

There is but one criterion and it is to be measured both real time and deferred time. It has been indicated that the first deferred time verification (see Figure 7.1.1(b)), which takes place while the lines are dry, will require a simulative signal to drive the valves. There is an obvious problem here, however. The valve output is modulated fuel flow (which is its output signal). During this verification, there will be no fuel, i.e., no

output signal. The solution to this problem in many electronic situations would be to inject a signal which could be altered or modulated. In this example, this would imply injecting a fluid under pressure to verify valve operation. This is a poor solution in this application and an alternative must be found.

One alternative to measuring output signal is to determine physical valve position. Will this give the same indication as an actual fluid measurement? This depends almost entirely upon valve construction. If the valve is constructed such that it is extremely unlikely for a failure to cause a disparity between a physically open valve and the open position of the actuator, then, signal measurements and valve position measurements should be approximately the same. It will be assumed that this is the case and position sensors will be used to determine whether the valve is open or closed. These sensors could be snap-action switches or reed relays.

The preceding solves the dry verification problem, and, as indicated below, improves the outlook of the remaining deferred time verification performed after fueling.

Using physical valve position solves one major postfueling verification problem identified earlier under Determination of Group Deduction Schemes. With this scheme, it will be possible to determine if one of the valves opens (in particular, the down stream valves). This greatly increases confidence in the results, but there is still no method of detecting a valve locked in the closed position without exercising the network. This cannot be done and the risk of a valve locking in a closed position after the last verification will simply have to be accepted. If the valves are reliable, this risk may be quite reasonable. In any event confidence in determining an open valve has been increased.

Since valve position is being used for deferred time verification, there is an additional feature which should be considered. Physical valve position will not necessarily detect a leaking valve. The postfueling verification is the only point prior to launch when such a check can be made. Therefore, in addition to valve physical position, a fuel detector should be placed in the common output line. This single detector is probably all that is necessary but confidence can be increased by adding two more – one between each pair of valves. Make-up of the fuel detector depends almost completely on the type of fuel. Several kinds of physical effects can form the basis for such a detector. Resistivity, pH and temperature are three examples. A fourth effect should be plausible in most cases and would likely be the most sensitive to minor leakage. This would be the addition of radioactive tracers to the fuel. In terms of redundancy verification methodology, the radioactive tracer would be considered a symbiotic signal.

The deferred time verification approach has now been determined. Attention can now be turned to the real time verification problem. As indicated previously, nonlinearity of the valves will prevent real time verification since they cannot be exercised. It might be possible to open the valves one at a time and verify their position. This would exercise each valve and increase verification confidence. Unfortunately, this would not increase the crew's confidence in vehicle safety and cannot be used. The only solution to approaching real time verification is an injected signal. This is the subject of the next design activity.

Determine Requirements for Real Time Injected Signals BLOCK E.3

To realize real time verification, an injected signal must exercise the valves and yet not interfere with the vehicle mission. These two requirements completely conflict and cannot both be met. It must be concluded that *real time verification during flight cannot be achieved.* The next best thing is deferred time verification which can be implemented by periodically burning the motor. This is not the best of all worlds but it does provide confidence in the verification results. How frequently the verification is to be performed depends on the fuel reserve and the desired confidence in status indications.

Develop Measures of Performance BLOCK E.4

The first task to be accomplished in this activity is to extract real time information which is indicative of the performance criterion. There are numerous methods of achieving this information, depending on the physical properties of the fuel. All of the fluid property measurements, however, suffer from the same shortcomings as flow and pressure discussed earlier under Determination of Group Deduction Schemes. In addition, most of these measurements require modifications of the fuel lines and possibly the insertion of sensors directly into the fuel flow. As a result, it seems quite feasible to use the same scheme designed for the initial deferred time verifications, viz., valve position.

The second step in this task is to establish a reference. Examining the performance criterion, it is obvious that valve position is not a deterministic signal which can be compared to an absolute reference value. Taken alone, valve position is stochastic, however, definite statements *can* be made about this signal relative to the input signal. If the input command is "close", the valve should be closed, and conversely. This is then an output/input reference situation.

Suppose the input command had the following characteristics:

+6 Vdc for open

0 Vdc for closed

If the valve position switches were supplied with 6 Vdc, a direct subtraction could be performed to determine relative valve position. This would be an extremely simple case of the inverse transform technique.

Since a direct subtraction is all that is required to provide a deterministic performance measure, special operations will not be required. Also, unless the environment is extremely noisy, the large signals, along with their step behavior, will probably not require smoothing.

Recall that status must be determined based on valve failure mode. This determination can easily be implemented into the mapping operation, the greatest problem being a more complex mapper than would ordinarily be required for a four element group.

7.1.4 Summary

A summary of the network redundancy features and the verification design features is presented below.

SUMMARY OF REDUNDANCY FEATURES

Features Not Favorable to Verification:

- Outputs (fluid dynamics) from redundant valves may not be individually accessible.
- The valve outputs are not isolated from the valves downstream.
- Valves are nonlinear.

Features Favorable to Verification:

- All valves are commanded simultaneously to the same position.
- Simple reference implementation.

SUMMARY OF VERIFICATION GUIDELINES

Recommended:

- Provide a passive sensor at each valve such that position can be sensed.
- Provide a fuel detector to check for leaking valves before launch.
- Use reliable valves since real time verification cannot be achieved.
- Determine status based on valve failure mode.

Not Recommended:

- Use of logical deduction.
- Use of a single sensor at the network output.
- Use of fluid property measurements.

7.2 EXAMPLE 2 – TWO-OUT-OF-THREE VOTING NETWORK (MAJORITY VOTE)

Description

This network consists of three channels and a voter. Each channel consists of a gyro and a rate switch which outputs a bilevel signal depending on whether the gyro rate output exceeds a set limit. This signal is the channel output which is provided to a

voter. Figure 7.2 is a simplified block diagram of the network. Each channel in the network monitors the same axis and axis rate is considered within bounds if at least two of the three channels indicate an in-bound condition. Each channel consists of identical components and all channels are active.

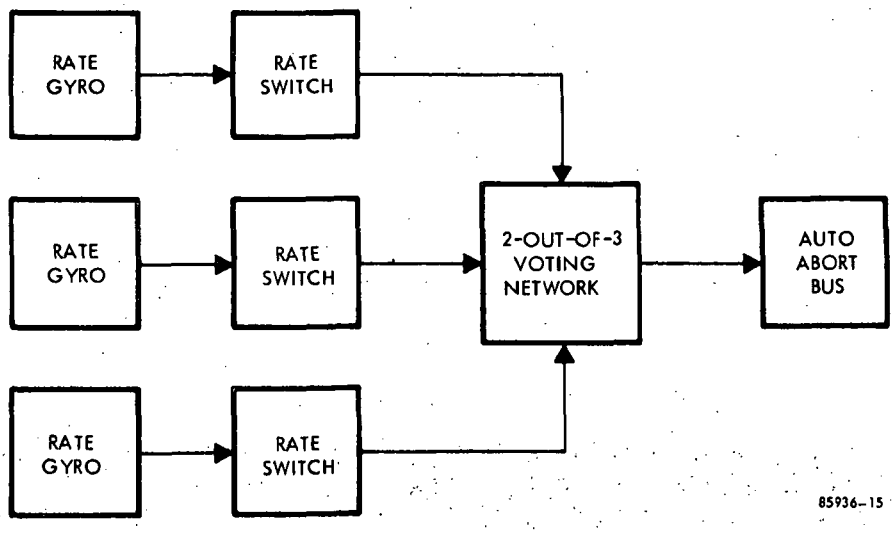


Figure 7.2. Simplified Functional Block Diagram of 2-out-of-3 Voting Network

The network incorporates a built-in self test capability for the gyros. The self test feature essentially consists of applying a known dc voltage to the gyro which will result in a predictable ac voltage output if the gyro is operating properly. This feature, *per se*, is a means of verifying gyro operation.

Critique

From the description above, the network can be identified as a simple set with each channel constituting an element of the set. Since voting is being used, redundancy verification can use the voter results (if accessible) for status indications. Depending on the voter design, these results may or may not indicate *which* element had become inoperative. In any event, the scheme will only detect status changes on an element basis, i.e., it will not be possible to tell if the rate gyro or the switch caused the change.

Using voter results will allow real time verification of the network. The extent of the verification, however, may be limited by a potential weakness in the basic approach. If the flight began with one of the three channels indicating an out-of-bound condition (due to failure within the channel), the voter may not recognize this condition. It may then be possible to begin a flight with one inoperative channel. Occurrence of this event should be avoided.

To be effective, the network must have a mechanism of independently verifying each channel prior to launch. This amounts to deferred time verification and has already been incorporated into the network as a gyro test signal.

The network has a drawback in that if two channels fail in a mode such that an out-of-bound condition cannot be indicated, the auto abort bus will not be notified of a true out of bound condition. Since this condition can only be checked on a deferred time basis, the risk must be accepted.

SUMMARY OF REDUNDANCY FEATURES

Features Not Favorable to Verification:

- Voter results may not be accessible.

Features Favorable to Verification:

- Redundancy verification is already built in.
- The level of verification is appropriate.
- Each channel is processing the same information.
- Voter operates on binary information.
- Incorporates a self test feature for each element to effect deferred time verification.

SUMMARY OF VERIFICATION GUIDELINES

Recommended:

- Use existing voter results.
- Use self test feature of the gyros to effect deferred time verification prior to launch.

Not Recommended:

- Attempting to duplicate the voter for verification purposes.

7.3 EXAMPLE 3 – SWITCHED REDUNDANT SET USING SPARE CHANNEL IN HOT STANDBY

Description

The network consists of three identical channels -- reference, command and spare. The reference and command channels function as a pair with the spare channel available in the event that a failure occurs in either the reference or command channels. During normal operation, the reference and command channels are constantly monitored by the comparator in the circuit. As long as the reference and command channel compare within a given tolerance, the command channel is used to provide the angular rate input to the flight control computer. However, if a failure occurs in either the reference or command channels, the comparator will sense an imbalance and cause the command

channel to be switched out and the spare channel to be switched in. The spare channel is not monitored prior to switching and is assumed to be functioning properly. Once the spare channel is switched into the circuit, it is not possible to remove it.

Figure 7.3 is a simplified block diagram depicting the manner in which the reference, command and spare channels are used to provide signals to the flight control computer. All three gyros monitor the same axis. The gyros are the same devices described in Example 2 and can therefore be individually verified as part of deferred time verification.

Critique

From the above description, the reference and command channels can be considered a simple set of degree one using compare-two as a real time verification technique. Accordingly, each channel is considered to be an element.

Since the compare-two technique cannot determine which element failed, a disagreement between the two channels is used to activate a switch and take both elements off line. At the same time, a third, identical element is switched on line.

The overall technique suffers from the classic case of verifying stochastic analog signals but appears to be a good solution to a particularly "sticky" problem. The approach is more reliable than attempting to majority vote analog signals but has the drawback of not indicating status of the spare channel. After launch, the vehicle is committed and there would be little the crew could do if it *did* know status of the spare channel. Status indications *would* be valuable, however, in determining the readiness of a vehicle to commit to translunar flight.

Status information derived from the network in Example 1 could be used to some degree to improve confidence in the results. This may be sufficient to allow a manual override of the no switchback condition.

SUMMARY OF REDUNDANCY FEATURES

Features Not Favorable to Verification:

- Spare channel not verified.
- Appears to be no means of implementing a symbiotic signal or for employing an inverse transformer technique.

Features Favorable to Verification:

- Partial redundancy verification is already built in.
- The spare channel is processing the same information as the command channel.
- Channel outputs should be identical.

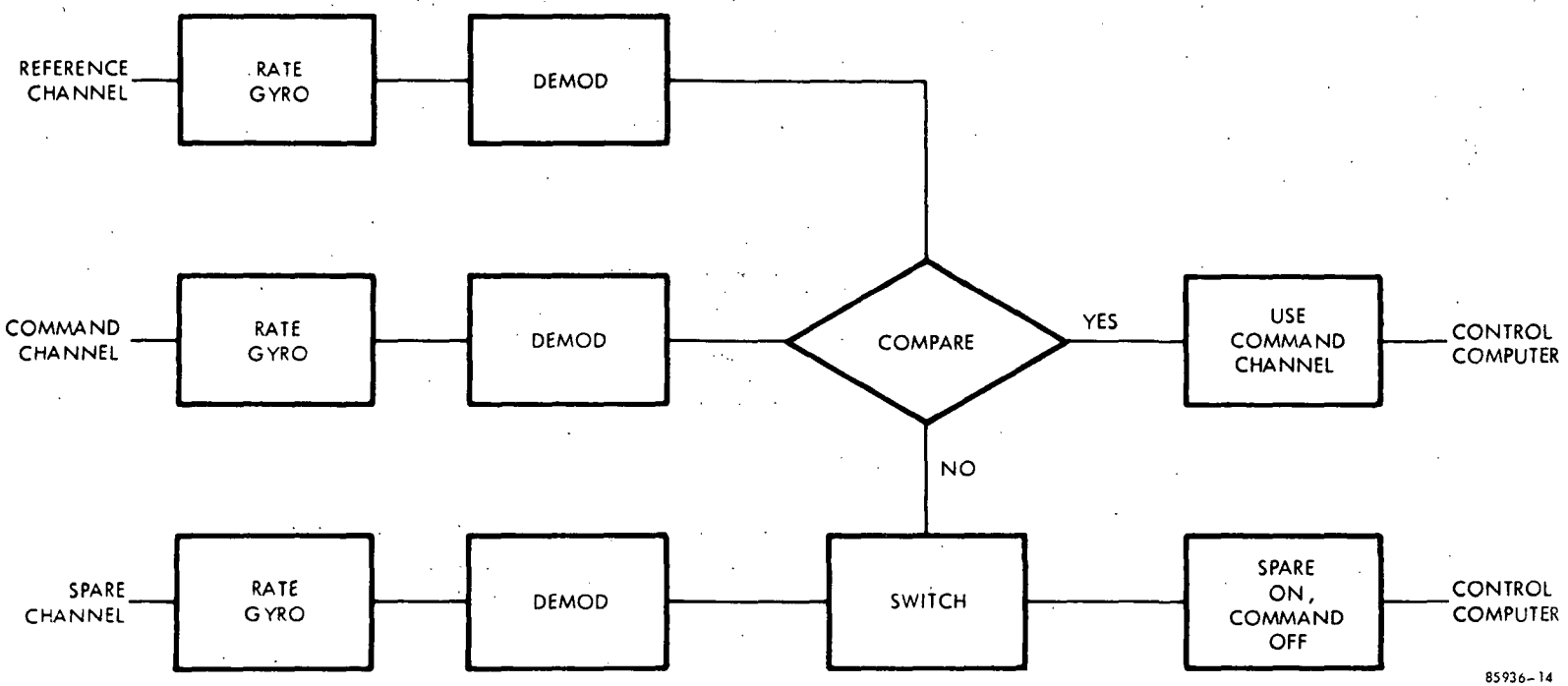


Figure 7.3. Simplified Block Diagram of Spare Channel in Hot Standby

SUMMARY OF VERIFICATION GUIDELINES

Recommended:

- Attempt to identify a means of employing a symbiotic signal.

Not Recommended:

- Verification below the channel level.
- Duplicating the existing verification equipment.

7.4 EXAMPLE 4 – ANALOG AMPLIFIERS WITH OUTPUTS HARDWIRED

Description

This network consists of two analog amplifiers with their respective outputs hardwired together. Each amplifier has identical characteristics and is processing the same input signal. The network is not maintained and both elements are active with no provision for switching between the elements.

Critique

This type of arrangement provides a continued degree of acceptable performance in the case where one of the amplifiers fails in an open condition. If at least one of the amplifiers fails in a mode where its output is shorted to ground, then the *network* output would go to ground potential. In any event, observing the network output to determine discernible variations indicative of the number of operative amplifiers would not be a practical means of deducing amplifier or network status. Determining the status of each amplifier by operating on the individual outputs appears to be the most direct method of verification and also provides the most information about the network.

Each amplifier output should be identical for any point in time hence negating any need for verification techniques which rely on timing information. The tenant signal can be used for verification purposes. It is desirable to use the current associated with the tenant signal, as opposed to voltage, since this network is not voltage distinguishable. On the other hand, the current will generally increase at the good amplifier if the other amplifier fails in a short condition. That is, a current change at an amplifier output could be indicative of a failure in that amplifier or the result of the other amplifier failing. A current change would also appear if there was a variation in the network load. If these variations can be minimized, current sensing may well be a good approach. It should also be noted that the technique of current sensing may fall short in detecting phase distortion on spectrum information.

SUMMARY OF REDUNDANCY FEATURES

Features Not Favorable to Verification:

- Network output will likely not vary according to the number of operating amplifiers.

- Each amplifier output is in the network throughput path and can be distinguished only under applicability of current sensors.
- Current sensing is not necessarily immune to element failure dependencies.

Features Favorable to Verification:

- Each amplifier is current-sense distinguishable from the other.
- Current at the output of each amplifier should be identical within a given tolerance.
- Both amplifiers are processing the same signal at any point in time.

SUMMARY OF VERIFICATION GUIDELINES

Recommended:

- Determine the status of each amplifier (if possible) with the use of a current-sensing device at the output of each amplifier.

Not Recommended:

- The use of isolation devices such as diodes at the output of the individual amplifiers since this practice would compound the verification process.

SECTION 8.0

OVERALL REDUNDANCY/REDUNDANCY VERIFICATION DESIGN GUIDELINES

8.0 OVERALL REDUNDANCY/REDUNDANCY VERIFICATION DESIGN GUIDELINES

This section is intended to place in capsule form the pertinent guidelines and "rules of thumb" identified in previous sections of the handbook. In addition, general statements regarding "do's and don'ts" for design practices are presented.

The information is presented in the form of a general statement backed up by assumptions, qualifications and cautions. The presentation has been styled such that each statement will have the following format:

Pertinent Statement
Assumptions: Reasoning: Cautions:

The practicality of applying real time verification to items being verified (IBV's) with failure rates less than 70 x 10^{-6} failures per hour should be carefully considered. As an index, this includes hardware less complex than most PC boards or standard rack-mounted power supplies.
Assumptions: The lower bound on the failure rate of a single instance of verification is 8 x 10^{-6}. **Reasoning:** Reliability of verification equipment should be approximately one order of magnitude better than that of the IBV to reduce false status indications from verification equipment failures. **Cautions:** Time sharing of verification equipment can influence this estimate.

Operations considerations contribute at least as much as redundancy design to the feasibility of verification.

Assumptions:

None.

Reasoning:

Operations considerations determine whether real time or deferred time verification can be used. Also, operations considerations influence confidence requirements and failure notification time.

Cautions:

The statement itself is more of a caution than a design limitation or guideline. Operations considerations must be understood before performance criteria can be established.

Hardwired redundancy is very difficult to verify.

Assumptions:

a. Distinguishable outputs from the redundant items are not available.

b. Combined output from redundant items does not vary detectably with the number of properly functioning items.

Reasoning:

If assumption (a) is true, it is impossible to identify which item has failed. If assumption (b) is also true, the redundancy cannot be verified.

Cautions:

The outputs of redundant items may not be distinguishable in a voltage sense, but distinguishable in a current sense.

It is possible that, in order to verify redundancy of a set, the status of each element must be known.

Assumptions:

The set consists of different types of elements.

Reasoning:

If a set consists of different types of elements (as might be used in fall-back or fail operational, fail safe approaches), it is important to know which element has failed.

Cautions:

This statement can be extended to first level groups since such networks rarely contain like elements.

The use of automated verification to achieve fault isolation within a redundant path of a set is a questionable practice. Fault isolation within a redundant path is best accomplished on a manual basis.

Assumptions:

Automated verification exists for the complete redundant path.

Reasoning:

a. Status of equipment *within* a redundant path of a set contributes no more information about redundancy status.

b. Determining status of additional equipment increases the likelihood of an overall verification error.

c. The added status information increases the likelihood of ambiguities in status resolution.

Cautions:

If the redundant path is quite large, automated verification within the path may be a reasonable compromise.

Sets are easier to verify than first level groups.

Assumptions:

a. The elements are of the same technology and complexity.

b. The set consists of identical elements.

Reasoning:

Groups will require deduction and sets do not. Groups will typically consist of several different kinds of elements. Verification equipment is more readily shared if elements are alike. Sets may have but one element on line; groups seldom do.

Cautions:

Exhaustive deduction on a first level group could be no more complex than that of a set consisting of different elements.

As a bound on the number of verification points, it is reasonable to assume this number will not exceed the number of elements in the principal system or be less than the number of first level groups.

Assumptions:

Verification will not be performed within an element; even if this element contains redundancy.

Reasoning:

Each first level group must have a verification point. Therefore, the number of verification points cannot be less than the number of first level groups.

Cautions:

a. The upper bound should be used advisedly since it is possible to have the number of verification points equal to the number of first level groups plus one.

b. A requirement for automatic reconfiguration can force the number to its upper bound.

c. A principal system which contains no first level groups could have only one verification point.

Real time verification will seldom yield confidence in status indications as high as deferred time verification which uses injected signals.

Assumptions:

The comparison is made on the same IBV.

Reasoning:

a. It is seldom possible to exercise an IBV over its operating range during real time verification. Also, real time verification can seldom examine all performance parameters of interest. Deferred time verification can be designed to do both.

b. If an error is observed during deferred time verification using injected signals, the verification process can be repeated. This is not possible for real time verification.

Cautions:

a. The deferred time verification described requires interruption of the IBV mission.

b. Extensive deferred time verification, even though performed on an automated basis, can be time consuming for large systems.

Redundancy can always be verified using deferred time verification, given it is possible to verify redundancy (although at a lower confidence) in real time.

Assumptions:

The comparison is made on the same IBV under identical operations requirements.

Reasoning:

Restrictions on real time verification are more severe than those of deferred time. If an IBV and associated operations requirements are such that real time verification can be employed, all tests for applicability of deferred time are automatically met.

Cautions:

Achieving increased confidence using deferred time verification usually requires that the principal system mission be interrupted while verification is being performed.

There is no restriction in the methodology which precludes the use of different verification schemes for each element in a set.

Assumptions:

Redundant elements in a set are off line.

Reasoning:

The methodology centers around the concept of individual IBV's. The only restrictions on the use of a verification scheme result from the fiscal and technical make-up of the individual case at hand.

Cautions:

Whether the use of different verification schemes on a set is good practice or not depends on the individual circumstances.

Using the proper reference technique contributes heavily to the feasibility of automated verification.

Assumptions:

None.

Reasoning:

a. For the eleven reference techniques identified, the cost can range over an 8:1 span.

b. Some of the reference techniques cannot be used for real time verification when the property of interest in the tenant signal is not deterministic.

Cautions:

Selection of a reference technique depends on the identified performance criteria. The statement can then be extended to include selection of performance criteria.

The smoothing operation is the key to control of status errors for real time verification.

Assumptions:

The mapping threshold cannot be altered but smoothing parameters can.

Reasoning:

The smoothing operation is intended to reduce random or unwanted fluctuations in the performance measure.

Cautions:

Increasing the amount of smoothing (and thus decreasing susceptibility to error) increases the response time of the smoother. This increases the delay between an actual status change and the indication of that change.

Decreasing the number of IBV's decreases the probability of an overall verification error.

Assumptions:

If deferred time verification is considered, the interval between verifications remains unchanged.

Reasoning:

Decreasing the number of IBV's decreases the number of decisions to be made.

Cautions:

Minimizing verification error may not be the only consideration. If the number of IBV's is decreased at the expense of introducing complex group deduction schemes, erroneous status indications due to verification equipment failure may outweigh the savings.

Actions taken to reduce the probability of Type I errors will adversely affect the probability of committing Type II errors.

Assumptions:

a. The comparison is made for the same decision process.

b. The variability of the status variable is fixed.

Reasoning:

Efforts to improve one or the other of these error probabilities ultimately result in altering the mapping threshold. This cannot be done without improving the probability of one type of error and decreasing the other.

Cautions:

If assumption (a) is not met, the statement does not necessarily hold.

Mapping thresholds (as implementations of status decision rules) must be set at constant values and must operate upon deterministic status variables.

Assumptions:

Achieving a variable threshold which is accurate is quite difficult.

Reasoning:

a. Mapping thresholds must be accurate. If requirements for variable references exist, these are best implemented before the smoothing operation.

b. A two level status mapper will frequently require two threshold devices. Achieving tracking on these two thresholds is a potential problem.

c. Performing variable reference operations before smoothing reduces the dynamic range of the smoothing operation.

d. A stochastic (nondeterministic) status variable will yield stochastic status indications.

Cautions:

Performing reference operations early in the status development operation results in requirements for handling low level signals. This decreases the overall noise immunity of verification.

Switches used in group iterative deduction schemes should have a probability of failure at least one order of magnitude less than the probability of failure for the set whose elements they are switching.

Assumptions:

The switches carry tenant signals.

Reasoning:

a. Since the reconfiguration switches carry tenant signals, they perform an on-line function and can cause a principal system failure.

b. Their contribution to principal system unreliability should be negligible.

Cautions:

a. Switches can often improve the reliability of a set over that achievable with hardwired redundancy. Therefore, the use of switches should not be considered routinely undesirable.

b. Switches will have three failure modes. Each should be considered in assessing the use of reconfiguration switches.

The necessity of status resolution can often be traded off for reference techniques which do not require resolution (but are often more complex).

Assumptions:

Regardless of malfunctions in preceding IBV's, the input to the IBV under consideration will remain admissible (although not necessarily of acceptable status).

Reasoning:

Reference techniques which depend on relative IBV operation (compare-two, inverse transform, voting) will usually indicate true IBV status even when the IBV input is not of acceptable status (but is still admissible).

Cautions:

a. Increased complexity of these reference techniques may outweigh any advantages realized by eliminating status resolution.

b. If there is a possibility that the IBV input can become inadmissible, status resolution may still be required.

Status resolution within a closed loop cannot be achieved without injected signals.

Assumptions:

The use of an injected signal is feasible.

Reasoning:

Successful operation of status resolution depends on the existence of serial or branched information flow so that sequential diagnosis may be applied. A closed loop does not provide this unless the loop is interrupted at some point by a signal whose status is known.

Cautions:

a. Real time verification within a loop will be difficult to achieve.

b. If injected signals cannot be used, the entire loop must be treated as an IBV.

GLOSSARY

Acknowledgment – Reference technique whereby the receipt and desired effect of a command is indicated through a separate return signal.

Admissible Input – An input which is within the range or repertory of an equipment such that it can be validly operated upon by the equipment transfer function.

Automated Redundancy Verification – A procedure whereby the presence (or absence) of redundancy can be verified without manual intervention at the redundancy network. This is not intended to imply that the verification procedure cannot be manually initiated, rather that manual reconfiguration (e.g., moving of cables) and checking (e.g., manual voltage measurements) at each redundancy network are not required. Redundancy verification implies that all elements in a redundancy network (both on-line and off-line) must be considered.

Coding – Reference technique wherein the performance measure is developed by determining the number of information errors or the rate at which they occur. This technique employs the characteristics of error detecting codes to establish a statement of status.

Compare-Two – Reference technique wherein like properties of two IBV outputs are compared to achieve an indication of their operational integrity.

Conditional Status – *See* Status, Conditional.

Correlation – Brief form for pseudo-random noise correlation. A reference technique wherein an IBV output is cross-correlated with a random input to determine IBV impulse response.

Cross-Power Spectral Analysis – Reference technique which involves the development of a coherence function resulting from the spectra of two IBV output signals.

Digital Signal – Any signal that is discrete in both time and signal space. During a finite time, a digital signal can send only a finite (or at most countably infinite) number of messages.

Element – The lowest level of equipment requiring status information. Each member or entity of a redundancy network is considered to be an element.

Error, Type I – The error which occurs when the verification equipment indicates a failure in the principal system when one has not occurred.

Error, Type II – The error which occurs when the verification equipment does not indicate a failure in the principal system within the notification time when one has occurred.

Failure, Type I A failure in the verification equipment which results in that equipment indicating a failure in the principal system when one has not occurred.

Failure, Type II – A failure in the verification equipment which will prevent that equipment from identifying a failure (within the notification time) in the principal system when one occurs.

Group – Any collection of elements which do not constitute a simple set.

IBV – Acronym for Item Being Verified. Associated with each IBV is a single verification point and status indication. An IBV can be an element, set or group. If a group is identified as an IBV, constituent element status must be deduced.

Injected Signal – *See* Signal, Injected.

Inverse Transform – Reference technique which performs, on the IBV output signal, an operation which is the inverse of that performed by the IBV and compares the result to IBV input.

Monitor Methods – General class of reference techniques which employ the storage of reference information concerning values of signal properties.

Notification Time – The delay time allotted for status development to indicate an IBV status change.

Off-Line Element – An element which is not communicating with successive functions. Depending on the circumstances, the element may or may not be performing its intended function, tenant signals may or may not be flowing and the element may be inert or stressed. Note that if the element is performing its intended function it will presumably be stressed.

On-Line Element – An element through which a tenant signal(s) is (are) flowing; the element is expected to perform its intended function and the output is communicating with successive functions.

Performance Criterion – A selected property (of possibly many properties) of an IBV which serves to provide an indication of performance used to make judgments about operational integrity. For example, an oscillator may use average frequency deviation over a prescribed period of time as a performance criterion.

Performance Measure – A verification signal which represents how well an IBV is performing with regard to a particular performance parameter. The performance measure provides an unsmoothed, automated indication of a performance criterion.

Principal System – The system which contains equipments whose status is to be identified. The baseline system which exists before addition of verification equipment.

Probability of Type I Error – The conditional probability that the verification equipment will commit a Type I error over the duration of the mission.

Probability of Type II Error – The conditional probability that the verification equipment will commit a Type II error.

Probability of Type I (Type II) Failure – The unconditional probability that the verification equipment will fail in a Type I (Type II) failure mode.

Redundancy – As used in this handbook, a term specifically meaning *functional redundancy*. The capacity of having more than one way to accomplish the same function where the alternative methods may assume the function within a prescribed time. So long as the alternate method is acceptable (in some sense), there is no implication of equivalent capability.

Redundancy Network – A general term describing a collection of functional entities (equipment, programs, etc.) which together form the primary and backup (or alternate) capability of performing a particular function. Each entity or member of the network is termed an element of the network. There is no restriction placed on the physical size or complexity of the network. A single nonredundant element would be a degenerate case of a redundancy network. This is redundancy of degree zero.

Set, Simple – *See* Simple Set.

Signal – The alteration of a physical property or properties (voltage, frequency, pressure, etc.), via a prearranged convention, in order to convey information. Whenever information is transferred, conveyed or related from one point to another, a signal is involved. For the purposes of redundancy verification, signals have been divided into three groups – tenant signals, injected signals and verification signals. From a mechanical viewpoint, the angular position of a gear or gyro, the longitudinal position of a translating member or the coordinate velocity of a vehicle are all signals.

Signal Form Analysis – Reference technique which determines signal property statistics and compares these with stored references of these statistics.

Signal, Idle – An injected signal which is used on-line in a real time basis to substitute for the tenant signal in cases where the tenant signal will be absent for an appreciable length of time. Idle signals are primarily used for a continuous verification policy under the above conditions. An idle signal should exercise elements of the principal system to the extent that the tenant signal exercises these elements and should not be the cause of any ambiguity with tenant signals. Idle signals will be primarily used in principal systems which remain quiescent over the major portion of their mission such as command destruct systems, sentinel alarms, infrequently used communications, etc. The use of an idle signal does not interrupt normal operation of the principal system.

Signal, Injected – A signal which has been added into the principal system for the purpose of verification. This signal can be static or dynamic. Injected signals can be further subdivided into idle signals, symbiotic signals and simulative signals.

Signal, Simulative – An injected signal which replaces the tenant signal during the interruption of normal principal system operation for the purpose of status identification. The simulative signal must exercise the elements through which it passes and is usually deterministic. Simulative signals are often called stimulus signals.

Signal, Symbiotic – An injected signal which is interlaced, multiplexed, mixed or in some fashion combined with the tenant signal on a noninterfering basis. Symbiotic signals are usually injected on a continuous basis and should exercise the elements through which they pass to the extent that the tenant signal exercises these elements.

Signal, Tenant – That signal which is inherent to the principal system – as opposed to any which are added for the purpose of redundancy verification. Tenant signals are those signals which exist in the principal system before any consideration is given to status identification.

Signal, Verification – Those signals which are inherent to the status identification equipment but are not injected signals. These signals consist of the command, control and general communication signals within the verification equipment.

Simple Set – A redundancy network whose elements all have the same predecessors and followers. Each element of the set is considered to be dedicated solely to the set's function and operating independently of other networks, i.e., not shared with other networks. Simple sets are often called "parallel redundancy." When there is no possibility of confusion, a simple set will be simply identified as a set.

Spectral Analysis – Reference technique which derives spectral characteristics of an IBV output signal. These characteristics are compared against a stored reference for each frequency band of interest.

Stand-Alone Set – A simple set which is not a member of a group. Stand-alone sets will always have a verification point at their output and either an injected signal or another verification point at their input.

Status – A qualitative and usually broad statement regarding the operational integrity of the item under contention, e.g., good, marginal, poor; go, no-go. The former is an example of a three level status indication and the latter, a two level indication.

Status, Conditional – Status determined from the condition of IBV output signal(s) without prior knowledge of the input signal *status* or *admissibility*. Unconditional status of the IBV can only be determined after the *status* of the input signal is known. For a two level status (i.e., acceptable or not acceptable), the following unconditional conclusions would be drawn from indications of conditional status.

- Input acceptable, output acceptable – IBV acceptable
- Input acceptable, output unacceptable – IBV unacceptable
- Input unacceptable, output acceptable – reserve judgment
- Input unacceptable, output unacceptable – reserve judgment

Status Development – The operation of extracting identified signal properties from an IBV and providing conditional status indications for that IBV.

Status Reporting – The operation of converting unconditional status signals to visual/audio presentations or making these signals available to additional equipment for further automated operations such as automatic reconfiguration.

Status Resolution – The process of identifying correct IBV status when several related IBV's all indicate an unacceptable condition due to a failure in one of the IBV's. Status resolution converts conditional status indications into unconditional indications.

Status Variable – A verification signal, either continuous or discrete, which is used for status decisions in the mapping operation. A status variable is a smoothed performance measure and represents the best estimate of how well an IBV is performing with regard to a particular performance criterion.

Value Check, Nonsequential – Reference technique which employs comparison of an IBV output signal property with a point reference value without regard to the order in which the signal property occurs.

Value Check, Sequential – Reference technique which employs comparison of an IBV output signal property in a sequential manner, deriving information both from the values and order of occurrence.

Voting – Reference technique whereby inference of operational integrity is drawn from a logical polling of outputs, or combinations of outputs, of three or more IBV's.

REFERENCES

[1] *Study for Development of a Handbook on Automated Redundancy Verification Techniques,* subtitle, "Design of Functional Redundancy." Phase II Report dated March 26, 1971 and performed under Contract No. NAS 10-7420.

[2] *Study for Development of a Handbook on Automated Redundancy Verification Techniques.* Phase I Report dated February 5, 1971 and performed under Contract No. NAS 10-7420.

[3] *Reference Data for Radio Engineers.* Howard W. Sams & Co., Inc.

[4] *Microwave Filters, Impedance Matching Networks and Coupling Structures.* Matthaei, Young, Jones.

[5] *Handbook of Filter Synthesis.* Zverev, A. I.

[6] *Smoothing, Forecasting and Prediction of Discrete Time Series.* Brown, R. G.

[7] *Statistical Theory of Signal Detection.* Helstrom, C. W.

[8] *Principles of Communication Engineering.* Wozencraft and Jacobs.

[9] *Random Signals and Noise.* Davenport and Root.

APPENDIX A

DISCUSSIONS OF REFERENCE FORMING TECHNIQUES

This appendix provides guidelines for selection of a reference technique. Since the choice of a reference technique is predominately influenced by performance criteria, these same guidelines can be used as an aid to determining the latter. Aside from physical restrictions of particular verification instances, the latitude of reference technique selections is probably the largest contributor to cost variability. This fact, together with the key role played by the reference in verification, dictates a complete treatment of the subject in this handbook. This appendix critically examines each reference technique and provides a summary of advantages and disadvantages at the end of each section. The techniques appear in alphabetical order.

A1.0 ACKNOWLEDGEMENT

Acknowledgement is an overall verification method and not truly a reference technique. It is being described for the sake of completeness since a verification approach which employs acknowledgement will not use a reference (in the sense of the other references described in this appendix). Since acknowledgement applies only to on-line equipment, the method qualifies as *redundancy* verification with the stipulation that backup equipment, as opposed to that primarily relied upon for the accomplishment of a function, is so reliable that status identification of only the primary equipment constitutes *redundancy* verification.* As such, the method is often applicable only to unsymmetrical redundancy.

The name in itself implies action. The concept is that equipment receives a command and, upon completing its assigned task, acknowledges that its duties have been dispatched. As might be suspected, the method finds its widest application in command and control systems. In fact, it is often the only means of verifying such systems. For maximum verification confidence, the functions of compliance and acknowledgement should be as closely associated as possible.

Consider a commandable switch which routes its output to either of points A or B (SPDT). Acknowledgement could be accomplished by (a), noting the appearance of the signal at A or B and transmitting notification of this fact as a return, or (b) using a separate contact to indicate switch position, perhaps being open when the switch is in position A and connecting to a 6 V reference when the switch is in position B. The second choice would probably be more reliable because sensing of the tenant signal would not be required.

Lack of accomplishment or lack of compliance would be the only failure types detectable by the technique. In most acknowledgement situations, Type I and Type II errors should occur only as the result of failure in verification equipment.

In some applications, an IBV failure will not be detected by this method until a command is initiated. Under these circumstances, serious consideration should be given to the use of an idle signal.

---

*If off-line elements in the principal system can be configured in a separate and parallel path, acknowledgement can be applied on each path. The method would then be a valid *redundancy* verification approach.

SUMMARY

Acknowledgement – Relative Advantages

- Knowledge of IBV input status not necessary. Input must be admissible, however.
- Generally simple and reliable
- Powerful verification method for command and control

Acknowledgement – Relative Disadvantages

- Idle signal often required for continuous real time verification
- Often applicable only to unsymmetrical redundancy
- Requires distinguishable outputs

A2.0 CODING TECHNIQUE

Coding techniques are basically methods contrived for the detection of errors in digital IBV's. The technique capitalizes on the power of Hamming codes or parity. In effect, the reference is carried along with each digital word in the tenant signal.* Probably the simplest example is the case of an IBV which outputs a digital data stream containing parity bits. Depending on the design of the system, some percent of message errors would be acceptable – say one percent. Then it is likely that, if a simple accounting system revealed that 20 percent of the messages being received contained errors, one would be willing to concede that a failure had occurred. Lest the obvious be overlooked, this also implies that coding techniques will detect *failures* (e.g., locked logic circuits) in equipment.

Since the technique utilizes information contained in the tenant signal for reference purposes, coding is obviously applicable to real time verification.

A statement of which, or even how many, types of failures are detectable through coding is difficult to make because of the strong dependence on principal system design and code selection. However, it may be pointed out that "failures" occurring between the points of encoding and error detection which do not result in message errors are likely of little interest.

Type I and II errors due to the verification equipment itself are both very unlikely with this technique (considering that a very clean signal will probably be provided as IBV output). Also, some tolerance to legitimate principal system errors is already built in. The principal contributor to Type I and II errors is the code itself. With careful design of the code, the probability of these errors can be made quite small.

Encoding and decoding equipment for these uses (especially for simple parity checking) is relatively simple and reliable.

It would be unlikely that unsymmetrical redundancy would influence the applicability of these techniques.

The sharing of coding verification equipment would generally be undesirable due to the necessity of dedicated error counting on any given IBV output. However, the cost of designing equipment to be dedicated to individual IBV outputs should be such that lack of sharability would be no cause for concern.

The coding technique will detect errors at the IBV output. The technique cannot determine if the errors occurred within or before the IBV. Therefore, this technique requires knowledge of input status in order to go beyond a statement of conditional status to a statement of status.

---

*This is the classic description of *block* codes. The technique is also applicable to convolution coding.

SUMMARY

Coding Technique – Relative Advantages

- Allows real time verification
- Simple implementations available
- Low likelihood of errors in statements of status
- Unlikely to be affected by unsymmetrical redundancy

Coding Technique – Relative Disadvantages

- Applicable only to digital systems
- Generally not compatible with concepts of shared verification equipment

A3.0 COMPARE-TWO TECHNIQUE

The compare-two class of reference techniques is one of the simpler approaches available and has the distinction, along with voting methods, of requiring at least redundancy of degree one. The technique is quite effective for real time verification.

The technique is generally capable of detecting a large number of failure types, the only requirement being that the failure result in a sufficient change in output from that which would be present from a properly operating element. The addition of excessive noise by the Item Being Verified (IBV) would be detectable by this method. Differentiation among failure modes is not possible when compare-two is employed. That is, a determination can be made that a failure has occurred, but the way in which an IBV has failed will remain in question. As such, the technique cannot indicate *which* element (in the set) experienced a failure.

Application of compare-two techniques may be on an analog basis using a simple differential amplifier or on a discrete basis with logical comparison done by computer or by specially designed, but probably uncomplicated, logic gates. The use of sampling implied by digital implementation of these techniques may not be sensitive to sampling rate. Since there may be no concern over signal information lost in the sampling process, sampling could be performed on a slow or even random basis as long as care is taken to assure the synchronization of sampling on the two element outputs under consideration. A further design option would be realized in determining a method for comparing sample values; the values could either be compared directly by differential circuitry, or could be encoded (perhaps to a binary form) and compared on that basis.

For the case of comparison on an analog basis, reference-forming equipment could be made very reliable, at least by comparison to implementations of more complex techniques. If a sampling approach were chosen, the reliability of the reference-forming equipment would be dependent upon the sampling equipment and the equipment performing the comparison of the samples. If the latter were, for example, a computer, the fact that it was being used for the comparison of two samples would not enhance its reliability beyond that for the case where the same computer was used for spectral analysis. The fact that lower sampling rates are a possibility in the compare-two technique might, however, allow the use of simpler, more reliable sampling equipment than that required by more comprehensive reference techniques.

In some cases, exploiting the simplicity of the compare-two technique will mean a sacrifice of accuracy. Whereas some methods allow comparison of an output property against a well-defined standard, comparison against another output forces the designer to allow for the possibility that both outputs will, at some time, vary within tolerable limits and in opposite directions. This could be crudely viewed as building twice as much "sloppiness" into the equipment.

Compare-two forms its reference evaluation on the basis of agreement between the two element outputs. In this regard, status resolution may not be required if the input to the two elements is admissible. That is, the transition from conditional status to status requires only the knowledge that the input is admissible – the status of the input is not required.

A restriction on these techniques is that, where the use of unsymmetrical redundancy means the use of redundant equipments with different capabilities, a comparison of their outputs may not furnish useful information. To realize its full potential, this class of techniques should be applied to "identical" elements, where "identical" is taken to mean alike within the practical restraints of equipment tolerances, noise environment, etc.

The compare-two reference technique may be used in time shared status development equipment. This may be envisioned in several different forms. For analog implementation, it would only be required that the output of the verification equipment be inhibited during the required switching among items to be verified – this provided that all outputs to be investigated by a given set of verification equipment be the same. In general, equipment doing the actual comparison would be sharable. Equipment preparing signals for comparison, such as sampling networks, would not be sharable.

Some degree of disadvantage must be attributed to this technique in that both IBV's must be powered up and operating. Where power conservation is desirable, as for example in a spacecraft, compare-two techniques will be less preferable than techniques which require only one powered-up element.

By comparison with other reference techniques, such as inverse transform and correlation, a minimum of design consideration with regard to the type of IBV and signal characteristics is demanded by compare-two techniques.

Sources of Type I errors in the compare-two technique will be lack of synchronization between two acceptable outputs, imbalance in signal property extraction devices and system noise. Viewing the signal comparisons here as the measurement of one element output against a "noisy reference signal" (the other element output) it appears that the likelihood of a Type I error due to noise will be somewhat greater than would result from a comparison against a noiseless reference. However, the aforementioned fact that any compare-two design would allow tolerance for variations in *both* element outputs would probably result in no net increase in Type I errors.

Type II errors will result from catastrophic failures in the reference forming equipment, the increased tolerance band and identical failures in the elements being verified. It should be noted that if IBV output signals are bilevel digital, the likelihood of identical failures may be greater since it is common for such logic to fail by locking up in one state or the other.

SUMMARY

Compare-Two – Relative Advantages

- Effective for real time verification
- Simple and, in some implementations, inexpensive
- Capable of detecting numerous types of failures
- Analog or digital implementations available
- Sampling rate, if used, not critical
- Computer implementation possible
- Requires only an admissible input to IBV

Compare-Two – Relative Disadvantages

- Not applicable to determination of element status
- Unable to distinguish among failure types
- Unable to identify which of the two elements failed
- May lack accuracy
- May experience limitations with regard to unsymmetrical redundancy
- Requires two powered-up elements
- Sensitive to low S/N situations

A4.0 CORRELATION TECHNIQUE

Correlation, as a reference technique, has been accepted to mean the cross-correlation of IBV output with a random input to achieve the IBV impulse response. This determined impulse response is then compared with a stored impulse response function (the reference). In general, this technique will be applicable only to linear, time invariant IBV's. Then, any elements or sets which involve encoding or decoding (including A/D and D/A conversion) as part of their functional operation are immediately excluded from consideration. Additionally, IBV's whose operation is not stationary in nature are also excluded from consideration.

In general, the applicability of correlation is not determined by the tenant signals, but by the nature of the operations performed by the principal system.

Since it is known the impulse response contains all observable information about IBV performance and since this information may be used to predict IBV response to any given input, such an analysis is indeed a powerful tool. Adding to the utility of this approach is the fact that a pseudo-random input may be injected at a level 10 to 30 dB below the tenant signal to be carried as a symbiotic signal. By cross-correlating this symbiotic signal (actually only a portion of the input signal) with the output, system impulse response may be obtained without disturbing information contained in the tenant signal.

In theory, correlation should be able to detect all failure types. This sweeping statement must be tempered with practicality, however. The actual capability of the technique to detect all failure types depends on the *extent* to which they influence the impulse response and the sensitivity of the remaining status determination equipment to detect changes in the compared functions.

As regards application to unsymmetrical redundancy, the choice of correlation as an approach is an acceptable one. In fact, the presence of unsymmetrical redundancy might allow the observation of set output while retaining the ability to identify the status of individual elements. This would be true whenever the elements had different transfer functions.

It will be desirable to accomplish correlation through the use of sampling techniques since the process requires the storage of signal values. No restriction on minimum sampling rate need exist; as a matter of fact, no requirement of periodicity need be placed on sampling – it may be done randomly. Accuracy will, however, depend on the number of sample points used to describe the impulse response and the point-for-point matching of the stored reference response.

The reliability of correlation equipment should rank low, along with the more sophisticated implementations of cross-power spectrum and inverse transform, because of its complexity as compared to that of compare-two and nonsequential value check.

Correlation can be used for real time verification but will not yield continuous results. For the technique to be effective, the IBV must be powered up, stabilized and operating in a single mode.

The greatest drawback to this technique is that the reference impulse response must be stored and compared on a computer. A special purpose machine which could be

time shared would be a more cost-effective choice than a general purpose machine. Also, a potential drawback is the requirement that the injected random noise be directly available to the correlator.

Random noise added to the signal by the IBV will be undetectable. This would probably be disadvantageous since, wherever the potential for the addition of a large amount of noise exists, it would probably be regarded as a failure type of interest.

Since a symbiotic signal is usually employed (a pseudo-random bit stream, for example), knowledge of input signal status will not be required to determine unconditional status.

The potential for Type I errors will be determined mainly by the design and equipment failures. For a given degree of accuracy in the generation of the impulse response, the degree of agreement which is demanded between generated and reference impulse responses will be determined by a trade-off between Type II errors and Type I errors. Due to its complexity, correlation will have a comparatively high probability of Type I error resulting from equipment failure or computation error.

SUMMARY

Correlation – Relative Advantages

- Allows real time verification
- Capable of detecting numerous types of failures
- Sampling rate not critical
- Adaptable to unsymmetrical redundancy
- May be made to be extremely comprehensive

Correlation – Relative Disadvantages

- Relatively complex implementation
- Use of sampling mandatory
- Design of accompanying special operations and smoothing strongly influenced by the kind of IBV
- Applicable only to linear, time invariant IBV's
- Not able to detect noise added by IBV
- Requires injection of pseudo-random signal

A5.0 CROSS-POWER SPECTRUM TECHNIQUE

The cross-power spectrum technique consists of taking the cross-spectrum of output pairs of the redundant elements in a simple set and forming a coherence function. This extracted coherence function is then compared to a stored reference value. Naturally, the technique is limited to frequency spectra information of analog signals.

The interpretation of the cross-spectrum is rather limited in scope without *a priori* knowledge of both the spectral content of the input signal and the element transfer function. Even this knowledge may not explain some high values of coherence at frequencies which were not expected to be coherent. A high cross-spectral measurement at some frequency may be attributed to a strong gain of the two elements at that frequency, a large input at that frequency, or strong contaminating noise common to both elements. Thus, the only feasible way of interpreting the coherence measure is by simply ascertaining that the coherence is high for any pair of outputs, over the frequency range of interest. The lower limit for "high" coherence will depend on the elements under consideration. Noise on the input to the elements will not degrade the coherence, since it should appear in the same form on the outputs of the elements. However, the noise added internally by each element will be incoherent with all components, and will thus degrade the overall coherence. Other factors which will affect the degree of coherence are the tolerances of the components in the elements and any external noise sources which affect one element more than the others.

Like other output/output reference methods (compare-two, voting), the cross-power spectrum technique requires that more than one element in a set be powered up and operating.

Any admissible input may be used, including the tenant signal or even random noise. Thus the S/N ratio of the input signal has no deleterious effect on the usefulness of the method. The cross-spectrum technique does not require dedication of principal system elements. A frequency limit is imposed by the type of sampler used, since the maximum frequency which can be resolved by the cross-spectral process (the Nyquist frequency) is one-half the sampling frequency. Any frequencies in the output above the Nyquist frequency should be removed with a low-pass filter to avoid aliasing error. Sampling rates of up to 10 MHz should be possible with current equipment.

The cross-spectrum technique is capable of indicating the *degree* of failure within an element. Out-of-tolerance failures result in reduced coherence between outputs, whereas loss-of-signal failures result in zero coherence.

Cross-power spectral analysis must be implemented on a computer and the coherence function calculated.

$$\overline{\alpha}^2 = \frac{\left|\overline{G_{xy}}\right|^2}{\overline{G_{xx}}\,\overline{G_{yy}}}$$

where

$\overline{\alpha}^2$ = coherence function

$\overline{G_{xy}}$ = average value of the cross-spectrum

$\overline{G_{xx}}$ = average value of the auto-spectrum for the first element

$\overline{G_{yy}}$ = average value of the auto-spectrum for the second element.

The technique's usefulness is diminished in cases of unsymmetrical redundancy comparisons, because the spectral content of the element outputs is likely to differ. The method may still be useful when the coherence suffers only small decreases.

The only equipment which must be located in close proximity to the IBV is the sampling and digitizing equipment. Once digitized, the sample data may be recorded or transmitted to the computer for analysis. No other requirements exist for location of equipment.

The data are sampled simultaneously from pairs of outputs, so time sharing of the sampling equipment is certainly possible. Sampling in pairs makes it necessary to power up two elements simultaneously for redundancy verification, but they need not remain on after sampling. Only two sampling and digitizing devices are necessary, and these can be manually or programmably switched to different elements.

The reliability of sampling equipment is generally very high, so the computer reliability limits the reliability of the verification equipment.

Type I errors will be primarily determined by failures in the sampling equipment. The most common failure is for one or more bits from the analog-to-digital converter to be locked in one state. This type error would be difficult to distinguish from prime system failures, because either one would degrade the coherence. Fortunately, sampling equipment is reliable and the likelihood of Type I errors will be small. Type II errors will be limited to failures and computation errors in the verification equipment.

In general, the cross-spectrum technique can be applied to any pair of elements for which the same spectral content is anticipated. It may also be used for indirect sensors, provided the outputs are expected to be the same.

SUMMARY

Cross-Power Spectrum – Relative Advantages

- The input need only be admissible
- Allows real time verification
- S/N ratio of input does not affect usefulness of method
- Any admissible signal, including tenant signal, may be used
- Time-sharing of sampling equipment is possible
- Provides comprehensive frequency domain information

Cross-Power Spectrum – Relative Disadvantages

- Limited to analog signals
- Requires first degree redundancy to detect an error; second degree redundancy to identify element status
- Technique is not as useful for unsymmetrical redundancies
- Necessary to power up two elements simultaneously
- Time and expense are greater than for many reference techniques

A6.0 INVERSE TRANSFORM TECHNIQUE

Inverse transforms, *per se,* are based on the mathematical premise that the product of an equipment transfer function and its inverse is unity. Then conceptually, a signal which is passed sequentially through an item having transfer function F(s) and one having transfer function $F(s)^{-1}$ will have been returned to its original form (input ≡ output). In effect, the IBV input forms the reference. The applicability, in theory if not in practice, of these techniques is dependent on the existence of a unique inverse transform and on the satisfaction of mathematical characteristics implicit in the statement above. First, the system (not the signal) must be stationary in nature. That is, its transfer function may not vary in time. In practice, it will be sufficient if the system is piecewise stationary and either is predictable in its time-variant behavior or is able to provide timely data on its own form. To see this, consider an amplifier whose gain is time-variable. Neglecting other parameters such as phase characteristics, if one knows the amplifier's (piecewise invariant) gain over a given span of time, a suitable amount of attenuation may be set up to provide the inverse operation. Knowledge of gain values may be had either by prediction or by the amplifier providing, as a separate output, gain information.

A second assumption implied by the use of frequency domain transfer functions is that the system is linear; i.e., the principle of superposition applies. If the time restrictions discussed above are met and the system is linear, a unique inverse transform will exist.

Inverse transform and inverse transfer function should not be confused. The above discussion of transfer functions is intended to illustrate the concept of inverse transform techniques. Actually, inverse transforms are more general than inverse transfer functions and will often exist when the latter does not. The transfer function theory above should then be used as an initial test of the applicability of this technique. As an example, consider the nonlinear operation,

$$e_{out} = m\, e_{in} + b$$

where m and b are constants. The inverse operation is described by

$$e_{in} = \frac{e_{out} - b}{m}$$

which is certainly a valid inverse transform. As an example of the applicability, consider the nonlinear operation for which a unique inverse does not exist,

$$e_{out} = k\, e_{in}^{2}$$

since a solution for e_{in} is

$$e_{in} = \pm\sqrt{e_{out}/k}$$

an ambiguity in sign exists. Still, if verification equipment could be given a rule for resolving the ambiguity, perhaps through the knowledge that e_{in} is always a positive quantity, inverse transform would be possible.

For a more practical example, consider an A/D converter. This is certainly a non-linear operation. The inverse operation would be performed by a D/A converter. In this case the problem in inversing arises from the fact that the reconstructed input signal suffers the effects of quantization errors. The acceptable degree of coincidence between this signal and the input signal which should be demanded must then be resolved.

A lack of unique inverse operation may be observed in the case of a limiter or clipper. A loss of information occurs in the operation which cannot be compensated for in the inverse operation. This is often complicated by the IBV not being exercised.

The inverse transform technique can be very comprehensive. With the assumption that all IBV failures of interest will be reflected as changes in that IBV's transform, the types of failure detectable become limited only by the accuracy of the reference-forming equipment. Generally, then this will be a very comprehensive approach to the problem.

The ability of verification equipment to distinguish among failure types is dependent on the method of comparison employed. If this comparison is by simple subtraction, no capability in this area will be realized. If comparison is by correlation, the potential for identifying failure types is great. (However, if correlation is to be involved, it might be more cost-effective to eliminate the inverse equipment and go to correlation techniques entirely.)

The inverse transform technique is capable of providing information on the conditional status of individual elements and will usually be applied to element output.

The verification equipment must be designed specifically to suit the equipment which it is to verify. This infers that a given set of verification equipment will be inflexible and only rarely will be sharable among sets.

Most implementations will require the comparison of input and inverse-transformed output without deletion of information from either. Therefore, if sampling is to be used on either input or output, the Nyquist sampling criteria should be observed. In effect, a lower bound is thereby set on sampling rate. It is unlikely that inverse operations performed on a sample set which cannot fully represent the output signal will lead to useful results.

At least two possibilities for implementation exist. One choice is the construction of dedicated equipment which operates real-time on the output signal. An example of this is an attenuator performing the inverse operation on amplifier output. A second choice is to digitize output signals and, by employing algorithms, perform the inverse operation. This may frequently be non-real-time as the input must be stored until completion of the inverse operation. Naturally, there would be quantization errors involved here. An advantage of the second choice would be that the storage of several inversing algorithms would render a computer sharable among sets.

An advantage of inverse transform over monitor methods is that only knowledge that the input is admissible will be required to transition from conditional status to status.

Additionally, the tenant signal can be used and nothing concerning its form or statistics need be known *a priori*. This may be a tremendous credit to this technique.

The potential of Type I errors will be higher when inverse transform is used than is the case when other techniques are employed. This is because unallowed-for inaccuracies

in verification equipment and almost all failures in equipment performing the inverse operation will result in a false indication of poor performance. In many instances the equipment performing the inversing will be complex in nature, costly, and comparatively (with respect to other candidate techniques) unreliable.

This technique is capable of performing real time verification. When the equipment performing the inverse operation is shared, the sharing would have to be on an automatic basis in order that verification qualify as continuous.

The likelihood of Type II errors will be determined almost entirely by the accuracy of the verification equipment.

Inverse Transform will not be a good choice for verification when element outputs which have been time multiplexed are to be observed since the multiplexing represents a nonlinear operation. In other situations, inverse transform equipment will not require timing information.

SUMMARY

Inverse Transform – Relative Advantages

- Effective for real time verification
- Capable of detecting numerous types of failures
- Computer implementation possible
- Does not require timing information
- Requires only admissible input
- Can be designed to distinguish among failure types
- Comprehensive
- Requires only one powered element at a time
- Can operate with tenant signals

Inverse Transform – Relative Disadvantages

- Little control over expense and complexity
- Restrictions on minimum sampling rate
- Design highly dependent on item being verified
- Verification equipment rarely sharable among sets
- Comparatively high potential for Type I errors
- Not applicable to element outputs which have been time multiplexed

A7.0 NONSEQUENTIAL VALUE CHECK TECHNIQUE

Being one of the monitoring methods, employment of nonsequential value check techniques entails the comparison of a signal property against a stored reference value. Obviously, then, one must be able to identify an acceptable range of values for the IBV output. In fact, the general requirement is that the signal under investigation be deterministic.

Implementations of this technique may be of continuous or discrete nature, both in time and value. For example, the direct comparison of a dc voltage against a fixed threshold is continuous in time and in value. Checking a digital word for its value would be discrete in time and value.

Since the technique involves comparison of IBV output to a stored reference, a special reference need not always be established. The reference operation can readily be implemented in the mapping threshold. This is considered a distinct advantage of this technique and is applicable so long as the reference is a constant value.

Value check techniques, like other monitor methods, are, in themselves, capable of giving only an indication of conditional status. To make a statement of IBV status requires knowledge of the *status* of the input. This requirement, along with that of having an input of deterministic nature, could severely limit the use of these techniques when it is undesirable to interrupt the principal system mission for verification. A requirement for deferred time verification is then a distinct possibility.

Because signal output is compared against a stored reference which is presumably very stable and noise free, the degree of accuracy achievable with value check techniques generally will be greater than that with those techniques which compare IBV outputs.

A relative disadvantage must be accorded to these techniques when it is desired to check values which are not constant in time, for then it is required that verification equipment be given timing information or the ability to derive such information. Even though sampling requires timing information, no restriction on minimum sampling rate is imposed as it is by spectral analysis. That is, there may be no interest here in compiling a sample set which completely describes the waveform. So, to the extent that the designer is satisfied with the level of confidence achieved, a signal value may be sampled for verification purposes at the rate of its occurrences or at any subharmonic thereof.

The reliability achievable in nonsequential value checking equipment will vary widely with the choice of implementation. Whenever synchronization capabilities are required, reliability of verification equipment will suffer. In the simplest case, such as that of comparing a dc voltage against a threshold, the technique can make use of mapping circuitry already in existence. Such circuitry is simple and reliable. More complicated and less reliable equipment would be required to check phase relationships.

Value check techniques are capable of detecting numerous failure types. But once again, the characteristics and capabilities are strongly dependent on signal property extraction methods, the relationship of the reference-forming equipment to any special operations performed and whether the reference can be implemented into the mapping operation. However, the general and significant statement may be made that nonsequential

value check techniques are not capable of distinguishing between or among types of failures. They are at most, able to point out that something is out of bounds.

Computer implementation of nonsequential value checking is a feasible approach. Encoded samples of signal values could be provided to a computer for comparison against a stored reference value. For such a simple operation, however, a small special purpose computer with values stored in read-only memories might be a more cost-effective choice than a general purpose machine.

Value checking equipment would be readily sharable among IBV's with identical performance requirements. If the value to be checked occurred only periodically in the IBV outputs, the choice facing the designer would be whether to sample all IBV outputs simultaneously and store the samples for sequential comparison against a stored value or to use no storage and dedicate the equipment performing the value comparison to one IBV output at a time. The first technique would require sampling; the second may, or may not, involve sampling.

As with other classes of verification techniques, flexibility in the location of reference-forming equipment will be dictated by the implementation chosen. If signal samples are encoded for comparison, transmission of these encoded samples can likely be accomplished with little sacrifice in verification confidence. For analog or nonencoded samples, the entire status development function would likely be in reasonable proximity to the IBV.

While the use of unsymmetrical redundancy may complicate the design of value checking equipment, the applicability of the technique will seldom, if ever, be negated by the fact that element outputs differ somewhat or by the fact that elements have different capabilities. The greatest impact of unsymmetrical redundancy on value check techniques would be in the case that it is desired to share verification equipment among element outputs. Here, differences in element output due to lack of symmetry could force the design and construction of additional verification equipment.

For the value check implementations requiring timing, an additional source of both Type I and Type II errors is provided. If timing information were incorrect but the signal actually good, a Type I error could occur; if both were incorrect, a Type II error could occur.

SUMMARY

Nonsequential Value Checks – Relative Advantages

- Simple implementations available
- More accurate than compare-two and voting techniques
- No severe restriction on minimum sampling rate
- Computer implementation possible
- Can be incorporated into the mapping operation

Nonsequential Value Checks – Relative Disadvantages

- Requires deterministic signal
- May require timing information
- Requires knowledge of input status for determination of equipment status
- May require deferred time verification
- Unable to distinguish among failure types

A8.0 SEQUENTIAL VALUE CHECK TECHNIQUE

The sequential value checking technique has a great deal in common with nonsequential value checking. Many of the statements made here also appear in the discussion of the nonsequential techniques. The major distinction between the two is that here, information concerning status is contained in signal values *and* in their order of occurrence. In nonsequential, order of occurrence is of no interest.

Sequential value checking is one of the monitor methods and, as such, carries the requirement of *a priori* signal knowledge. Indeed, a deterministic signal is required since verification equipment is expected to "know" what values are acceptable and in what order they should occur.

The fact that time-sequential values are to be observed implies a process that is discrete in time. Where discreteness in time is not a characteristic of the signal under observation, it can be established by quantization, sampling, or digital encoding. Comparison of signal values with stored values may be accomplished on a continuous or a discrete basis. The use of unquantized signal samples for comparison would be a continuous realization; digitized signal samples represent a discrete value comparison. An analog signal such as a sine wave would generally not be compared without sampling since an entire waveform, rather than discrete values, would have to be stored; a bilevel bit stream, on the other hand, would already be discrete in both time and value. An example of the latter would be the observation of a bit stream in order to determine the presence of a frame synchronization word.

Sequential value checking equipment will, in general, require timing information or synchronization capabilities since it must "know" when the sequence of values to be checked is in progress. Often the achievement of synchronization through the use of digital logic will be relatively simple.

One important application of sequential value check is the limit check. Here, the value of a signal property is measured at time t. This value is then compared to the value of the property at (t+T) and a decision made on the difference. This same principle can be used to verify that an event has occurred within some established time interval.

When element outputs are time multiplexed signals, sequential value checking will be a definite candidate. The serial bit stream will already contain synchronization information so that the establishment of timing should be a simple matter.

Like nonsequential value checks, this technique requires a knowledge of input status to allow the conversion of conditional status into status. As has been pointed out, this requirement along with that of a deterministic signal, will in some cases result in interruption of the principal system mission to effect verification. This implies deferred time verification implementation.

Value check methods will be inherently more accurate than those techniques which compare output signals against each other since here, one constituent of the comparison is a reference which can presumably be made very precise. The fact that sequential value checking will involve time in all cases adds one degree of freedom over many nonsequential applications. Thus, in general, sequential will be less accurate than nonsequential value checking.

As with the nonsequential case, no minimum is placed on sampling rate since there is often no concern over output signal information loss. If sampling is to be employed, sampling rate will be determined by the rate-of-occurrence of the values to be observed and the level of confidence to be demanded of the verification.

Equipment for comparing sequential signal values can be made simple and reliable. The ease with which timing is established will vary greatly with the situation. In some cases, timing will be available from principal system equipment; in another case, verification equipment may be asked to synchronize to a quantized form of the signal to be observed. Obviously the requirements of the second case may lead to the use of sophisticated equipment.

Sequential value checks can be given the ability to detect a large number of failure types. Since information is available from both the values and the order of their occurrence, the potential exists for this technique to be more comprehensive than nonsequential value checking. Only an investigation of failure modes of the principal system equipment will reveal whether the identification of failure types is possible by sequentially checking values. That is to say, if all failure modes affecteach value in the sequence in the same way, no discrimination among types of failures will be possible. The order-of-occurrence information, then, will be entirely involved in the establishment of verification confidence.

Sequential value check can be applied to status determination of both sets and elements, but the former application is quite restrictive.

Computer implementation, either by special or general purpose machine, is a definite candidate approach to value checking.

Sequential value checking equipment can frequently be shared among IBV's having identical performance requirements. The degree to which equipment would otherwise be sharable would likely be limited due to the different timing requirements which would generally exist between different IBV's and, less importantly, the requirement for storing additional sequences. If the sequential checking were implemented by logic gates and/or shift register, the design would be with consideration for the equipment to be verified and the signal to be observed. Their adaptability to other principal system equipments and signals would likely be small. If implementation were by computer, the machine could act simply as several sets of verification equipment, selecting the proper functions for the equipment and signal under scrutiny.

The general rule for flexibility in the location of equipment will be observed for these techniques; if signal samples are encoded for comparison, transmission of these encoded samples can likely be accomplished with little sacrifice in verification confidence. If nonencoded samples or analog values are chosen for comparison against reference values, this comparison should be performed before any transmission is attempted.

The use of unsymmetrical redundancy will usually not affect the applicability of sequential value checking, though it may complicate or negate the sharing of verification equipment among element outputs.

Since value checks may be performed on one element at a time, if elements require prime power, this power need be applied to only one element at a time for value checking. This is in contrast to requirements for compare-two and voting techniques.

The requirement of timing information, generally imposed by sequential value checking, should be viewed as an additional source of Type I errors since erroneous timing can cause a disagreement between signal values and stored values even though the signal is entirely proper. Of course, erroneous timing can also cause Type II errors, though the likelihood of such an error is remote. Also, attempts to share verification equipment among IBV's open the possibility of checking against the wrong set of stored values. Sequential value checking techniques are not tolerant of legitimate system errors. If a value does not check "good," the equipment has no way of determining whether the cause is a failure or an error. The effects of such legitimate errors, as well as those of noise, may be minimized through the smoothing operation.

SUMMARY

Sequential Value Checks – Relative Advantages

- Simple implementations available
- More accurate than compare-two and voting techniques
- No severe restriction on minimum sampling rate
- More comprehensive than nonsequential value checking
- Computer implementation possible

Sequential Value Checks – Relative Disadvantages

- Requires deterministic signal
- Requires timing information
- Usually not applicable to determination of set status
- Requires knowledge of input status for determination of equipment status
- May require deferred time verification
- Limited in ability to distinguish among failure types
- Timing requirements introduce additional source of errors

A9.0 SIGNAL FORM ANALYSIS TECHNIQUE

Signal form analysis techniques involve the derivation of signal statistics (rms value, mean value, variance) for comparison against a reference value.

The development of many signal form indicators such as rms value, mean value and variance, require time for integration or the collection of a sample set. This does not, however, negate the possibility of real time verification since such samples can often be collected very rapidly. If the signal statistics vary as functions of time, implementation of this technique can be greatly complicated or perhaps impractical.

The number of failure types detectable through the use of these techniques will be principally determined by the number of failure types which evidence themselves as variations in the signal statistics under observation – the rms value of an oscillator signal will not generally disclose frequency drift. The comprehensiveness permitted by these techniques will be determined only by analysis of the particular situation in which their application is contemplated.

Depending on the signal statistics to be investigated, implementation may be either on an analog or discrete basis. Mean and rms values of analog signals will be easily derived on an analog basis; the variance of a stochastic signal would more than likely be determined on the basis of a sample set. Sampling techniques will generally be applicable to the investigation of signal statistics. Usually, those properties which are amenable to analog implementations can be observed on a sampled basis. As a rule, minimum sampling rate will be determined by the signal statistics and/or by the requirements for verification confidence. Statistics such as mean, variance, and rms may be determined by sampling on a "slow" basis (slow by comparison to signal frequencies) or on a periodic or random basis.

These techniques involve both the process of developing a signal statistic and the comparison of this statistic against a stored reference. Both of these functions might be performed on a digital basis. As an example, a signal could be sampled and these samples encoded. A computer could then determine the mean of the sample set and compare the derived value against a stored reference value.

The reliability of the verification equipment for signal form analysis will vary over a wide range, depending on the signal property chosen for investigation and on the selection of a basis of comparison. By comparison to swept frequency mixing of spectral analysis, the use of these techniques will usually result in acceptable reliability of verification equipment.

In contrast to some other techniques, the inherent time delay of this technique will typically allow erroneous information to be transmitted before an unacceptable status indication is realized. This could be a drawback for large filter constants with critical information. Trading off shorter filter constants with the probability of error will inevitably result.

Equipment required to implement signal form analysis techniques could be shared among the elements of a set without complication. For identical elements both the equipment developing the signal statistic and that performing the comparison would be sharable.

The design of signal form analysis equipment would usually be specialized to the IBV and its signal and would, hence, likely be unsharable among nonidentical IBV's.

The likelihood of errors, both Type I and Type II, attributable to this technique would be determined almost exclusively by the sample size or integration time. Susceptibility to these errors will inevitably be traded off with status notification time.

SUMMARY

Signal Form Analysis – Relative Advantages

- The only effective monitor method for verifying stochastic signals in real time
- Computer implementation possible
- Requires one powered-up element
- May be no severe restriction on sampling rate

Signal Form Analysis – Relative Disadvantages

- Requires input status to allow transition from conditional status
- Usually unable to identify as many kinds of failures as techniques observing signal values or spectral characteristics

A10.0 SPECTRAL ANALYSIS TECHNIQUE

Spectral analysis techniques essentially involve the extraction of frequency information from IBV output signals in terms of energy in bands and compare these values with stored references. Implementations may take the form of swept frequency mixing, lumped constant filters and chirp filters. All these implementations are typically concerned with the investigation of a single signal. This signal may be a set output if element failures are reflected as changes in the spectral content of the set output. More often, spectral analysis will be applied to an element or group output.

As with all monitoring techniques, the requirement of *a priori* knowledge is present. Here, the requirement is placed on the frequency content of the signal under investigation. In many cases, the necessity of an output with known frequency content will mean that a known input will have to be provided. This implies a simulative signal and the use of deferred time verification.

Spectral analysis is capable of detecting a wide variety of failure types. Loss of signal, excessive noise, and signal distortions are all detectable by at least one of the implementations.

Spectral analysis using one or two lumped constant filters might be achievable rather simply. In certain instances, however, a bank of filters or a set of swept filters will be required. The effect of the latter on reliability, errors, maintenance and cost should be obvious.

The technique may be required to furnish a great deal of information. One could require simultaneously, for example, that third harmonic content be in a certain ratio to the fundamental, that the fundamental be within certain ranges of amplitude and frequency, that even harmonics be missing, etc. Each of these requirements would be treated as an individual signal property. The result would be several individual status development implementations and a multidimensional status variable.

Spectral analysis may be implemented using sampled data. This fact makes the technique amenable to time sharing among several identical IBV's. The sampling rate would be critical as the Nyquist sampling criterion would have to be observed and the fixing of a sampling rate would imply an assumption of fundamental frequency.

Spectral analysis is capable of real time verification, but many real-world signals will have neither constant nor predictable frequency content. The situation then would require the use of a simulative signal and deferred time verification.

It is true of spectral analysis, as it is of the other three monitoring techniques, that the transition form a statement of conditional status to one of status requires knowledge of input status. This must be considered less desirable than requiring only an admissible input as is the case with some other techniques.

The primary sources of Type I errors in these techniques are failures in the verification equipment and system noise.

Since this technique is exclusively analog in nature, all equipment necessary for the development of a status variable should be located in proximity to the IBV's in order to avoid noise pollution and signal distortion by transmission.

The technique finds particular applicability where a frequency-multiplexed signal is of concern – for example, on a data bus.

SUMMARY

Spectral Analysis – Relative Advantages

- Allows real time verification
- A practical frequency domain technique
- Does not require timing information
- Advantageous in low S/N ratio situations
- Adaptable to frequency multiplexed signals

Spectral Analysis – Relative Disadvantages

- Predominantly applicable to analog signals
- Minimum sampling rate requirement
- Requires knowledge of input status for transition from conditional status to status
- May require deferred time verification

A11.0 USER COMPLAINT TECHNIQUE

The user complaint technique, like acknowledgement, is an overall verification method and not truly a reference technique. It is being described for the sake of completeness as it is certainly not an automated means of verification. Since user complaint applies only to on-line equipment, the method qualifies as a *redundancy* verification technique with the stipulation that off-line equipment is extremely reliable.

The need for *a priori* knowledge is implied in that the user must have stored information on nominal performance against which to compare actual performance. This, then, is like a human-implemented monitored method. The failure types detectable will obviously be those which manifest themselves as output which the user can identify as erroneous.

Potential for errors, both Type I and Type II, is great because the performance criteria will often be quantitative and the mind is notoriously poor for accurately recalling quantitative matter. Also, the signals to which these techniques apply will be limited to those observable by a human.

Of course, the advantages of human capabilities, including the ability of handling unforeseen occurrences, will be realized. It is interesting to note that, ultimately, the user will always detect a failure; although the time lag could be significant in some cases. If a system exists wherein the user would not detect a failure, one must seriously question the utility of such a system. There will be some instances where user complaint, although not an automated technique, will be a sound engineering solution.

The application of this technique to personnel safety systems has obvious disadvantages and, in this one area, should be avoided.

SUMMARY

User Complaint – Relative Advantages

- No additional equipment design
- Inherent ability to cope with the unexpected
- No verification equipment cost
- Economical

User Complaint – Relative Disadvantages

- Not automated
- Typically applicable only to unsymmetrical redundancy
- Requires *a priori* knowledge in the form of user experience
- Requires distinguishable outputs
- High potential for errors
- Potentially long time delay to failure identification
- Not practical for personnel safety systems

A12.0 VOTING TECHNIQUE

It should be pointed out that the voting techniques to be discussed are voting techniques applied for the purpose of redundancy verification and do not *necessarily* have anything to do with the manner in which redundancy is achieved. Majority voting for the purposes of error elimination or the achievement of redundancy is of interest only to the extent that it provides an existing reference technique.

When used as a reference technique, it is characteristic of voting that no distinction can be made among failure types. That is, when the results of a polling procedure indicates a disagreement among element outputs, it is generally not possible to determine whether the cause of the disagreement is noise, loss of signal, signal distortion, etc.

Perhaps a strong advantage of the voting technique is that input status will usually not be required for a transition from conditional status to status. If indications are that two of three outputs agree or that all outputs agree, one will likely require only a knowledge of input admissibility before he is willing to pronounce element status.

While sampling rates would affect the confidence and accuracy of voting, there is no minimum placed on sampling rate as is the case with spectral analysis. It is conceivable that, for a clean digital bit stream, one would be satisfied to vote on one bit out of every ten, one-of-fifty, or one-of-one-hundred.

The voting technique is applicable only to distinguishable element outputs.

Because of the simplicity and inexpensiveness attributable to digital voting equipment, it would not likely be attractive to time-share voting equipment among IBV's.* An exception would be when computer implementation was used and then sharing should be a simple matter. Note that while digital implementation of a voting technique may be relatively simple, analog implementations, though possible are usually quite complex.

Type I errors may occur due to system noise or failures in reference-forming equipment (voter). Type II errors will be very unlikely in the reference-forming equipment so long as the redundancy network is not allowed to continue performing after the majority-less-one of the elements fails. For example, two elements could fail in the same way. If this happens in a triple redundant configuration, both Type I and Type II errors may occur: two failed IBV's would be identified as acceptable while the one good IBV would be identified as unacceptable. Similar failures in all three IBV's would result in an indication that full redundancy was present. The likelihood of such occurrences is not so remote if the IBV's contain digital logic. It is very common for such equipment to fail by "locking up" in one of its legitimate states. Also, total loss of signals from IBV's might result in an indication of unanimity.

To reduce errors in verification, the voter should be located in proximity to the IBV's. If samples of IBV output are digitized before voting, these digitized samples could be transmitted for voting elsewhere with no loss of accuracy.

---

*This statement is directed at the voter itself. If two additional IBV's had to be added to realize a reference by voting, the statement obviously does not apply.

Unsymmetrical redundancy (redundancy consisting of different element types) will affect the applicability of the voting technique only insofar as the IBV outputs differ. If IBV's perform their functions in different manners but have identical outputs, it is of no consequence. The degree of dissimilarity which can be accommodated by the voter without sacrificing verification confidence will be entirely dependent on the nature of the dissimilarity and the particular situation.

It is obvious that, if IBV's require prime power, all IBV's to be involved in the polling procedure must be powered up. This could be of concern if power consumption were critical or if an off-line equipment situation exists.

The voting technique would probably not be a good reference choice in low S/N ratio situations, since the effect of noise could cause apparent disagreements when, in actuality, all IBV's were operating properly.

As with compare-two, the fact that IBV outputs are being compared against each other (rather than against a stored reference) implies some sacrifice in accuracy for analog implementations. That is, allowance must be made for outputs to drift to extremes within their range of tolerance with no indication of disagreement.

SUMMARY

Voting Techniques – Relative Advantages

- Simple, reliable, and inexpensive voter implementations available for digital applications
- Effective for real time verification
- Can be made to identify element status
- Requires only admissible input for status determination
- Implementation by simple computational machines possible (digital)
- No restriction of minimum sampling rate

Voting Techniques – Relative Disadvantages

- Unable to distinguish among failure types
- Less accurate than comparison against a stored reference
- May be affected by unsymmetrical redundancy
- Requires all polling elements to be powered
- Not advantageous in low S/N ratio situations
- For a three element case, the techniques are unable to give a conclusive status indication when two failures occur.
- If an element must be added to a set to achieve the technique, the final design can be noncompetitive from a cost and utility standpoint.
- Analog implementations are considerably more complex than digital.

APPENDIX B

FILTER EVALUATIONS

[B1.0]

This appendix contains the low pass filter evaluations for Section 4.5.2 in the body of the handbook. Mathematical expressions are developed for pertinent filter characteristics used in smoothing analog signals. The following filters have been evaluated.

- Single Pole Filter
- Double Pole Filter
- Constant Time Integrator
- Ideal Rectangular Filter

B1.0 SINGLE POLE (EXPONENTIAL) FILTER

The single pole filter is a simple RC type low pass filter. The adjective, exponential, is due to its exponential weighting function or impulse response.

The impulse response, I(t), is

$$I(t) = \frac{1}{\tau} e^{-t/\tau}, \; t \geq 0 \qquad \text{(B1.0)}$$

where τ is the RC circuit time constant. The maximum value of I(t) occurs at t = 0 and is, $I_{max} = \frac{1}{\tau}$.

The frequency response is the Fourier transform of Equation (B1.0)

$$H(\omega) = \frac{1}{1 + j\omega\tau}$$

The power spectrum or spectral density is

$$|H(\omega)|^2 = \frac{1}{1 + (\omega\tau)^2}$$

The single sided noise equivalent bandwidth, B_{eq}, is defined as

$$B_{eq} = \frac{1}{2\pi}\int_0^\infty |H(\omega)|^2 \, d\omega$$

For the single pole filter,

$$B_{eq} = \frac{1}{2\pi}\int_0^\infty \frac{d\omega}{1 + (\omega\tau)^2} = \frac{1}{4\tau} \text{ Hz}$$

Also, $\tau = 1/(4 B_{eq})$

The step response, R(t), is

$$R(t) = 1\text{-}e^{-t/\tau}$$

In terms of B_{eq}, the frequency spectrum is

$$|H(f)|^2 = \frac{1}{1 + \left(\frac{\pi}{2} \frac{f}{B_{eq}}\right)^2}$$

Also, $I_{max} = 4B_{eq} = 1/\tau$

$$R(t) = 1 - e^{-4B_{eq}t}$$

Rise time is defined as that value of time (t_r) such that the step response is,

$$R(t_r) = 0.9$$

and rise time is

$$t_r = 0.572/B_{eq}$$

B2.0 TWO POLE FILTER

The general frequency response equation for the two pole filter is

$$H(\omega) = \frac{1}{1 - \left(\frac{\omega}{\omega_n}\right)^2 + 2j\, z\, \frac{\omega}{\omega_n}}$$

where z is the damping coefficient.

The spectral density is

$$|H(\omega)|^2 = \frac{1}{\left[1 - \left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\, z\, \frac{\omega}{\omega_n}\right]^2} \qquad \text{(B2.0-1)}$$

For overdamped cases (z<1),

$$I(t) = \frac{\omega_n}{\sqrt{1 - z^2}}\, e^{-z\omega_n t}\, \sin\left(\sqrt{1 - z^2}\,\omega_n t\right)$$

and

$$R(t) = 1 - \left[\cos\left(\sqrt{1 - z^2}\,\omega_n t\right) + \frac{z}{\sqrt{1 - z^2}} \sin\left(\sqrt{1 - z^2}\,\omega_n t\right)\right] e^{-z\omega_n t}$$

The overdamped case of $z = \sqrt{3}/2$ realizes a fast rise time circuit without undue ringing. Using this value in Equation B2.0-1 and performing the indicated integration,

$$B_{eq} = \frac{1}{2\pi} \int_0^\infty |H(\omega)|^2 d\omega = \frac{\omega_n}{7} \quad Hz$$

Substituting $\omega_n = 7\, B_{eq}$ and $z = \sqrt{3}/2$ into Equation B2.0-1,

$$|H(f)|^2 = \frac{1}{\left[1 - \left(\frac{2\pi f}{7\, B_{eq}}\right)^2\right]^2 + 3\left(\frac{2\pi f}{7\, B_{eq}}\right)^2}$$

Then

$$I(t) = 14\, B_{eq} \exp\left(-7\ \sqrt{3}/2\ \ B_{eq} t\right)\ \sin\ \left(7/2\ B_{eq} t\right)$$

I (t) is maximum at $B_{eq} t = \pi/21$ and $I_{max} = 2.826\, B_{eq}$

Finally,

$$R(t) = 1 - \left[\cos\left(\frac{7}{2} B_{eq} t\right) + \sqrt{3} \sin\left(\frac{7}{2} B_{eq} t\right)\right] \exp\left[-7\sqrt{\frac{3}{2}}\ B_{eq} t\right] \quad \text{(B2.0-2)}$$

These latter expressions, with $z = \sqrt{3}/2$, are the ones used to represent the double pole filter. A numerical solution to Equation B2.0-2 yields a rise time,

$$t_r = 0.465/B_{eq}$$

APPLICATION NOTE: Both single and double pole filters can be made of simple passive devices down to about B_{eq} = 100 Hz. Using modern FET operational amplifiers, filters can be implemented with B_{eq} less than 0.01 Hz. This is expected to be adequate for any application.

B3.0 CONSTANT TIME INTEGRATOR

The constant time integrator has long been a candidate for smoothing filters. This device basically performs the operation

$$0(t) = \frac{1}{T} \int_{t-T}^{t} f(x)\ dx$$

Where 0(t) is the output at time t, and f(x) is the input function of time. T is the duration of the integration. The inpulse response is 1/T whenever t = 0 falls within the limits of integration and zero otherwise.

$I(t) = 0;\ t < 0, t > T$
$I(t) = 1/T,\ 0 \leq t \leq T$.

Likewise, if f(t) is the unit step function,

$R(t) = 0,\ t < 0$
$R(t) = t/T,\ 0 \leq t \leq T$
$R(t) = 1,\ t > T$.

The rise time is given by

$R(t_r) = 0.9$
$t_r = 0.9T$

The frequency response is the Fourier transform of the impulse response

$$H(\omega) = \int_o^T \frac{e^{-j\omega t}}{T}\, dt$$

$$= e^{-j\omega T/2} \left(\frac{\sin \omega T/2}{\omega T/2} \right) .$$

The power spectrum is

$$|H(\omega)|^2 = \frac{\mathrm{Sin}^2\,(\omega T/2)}{(\omega T/2)^2} .$$

$$B_{eq} = \frac{1}{2\pi} \int_o^{\infty} \frac{\sin^2 (\omega T/2)}{(\omega T/2)^2}\, d\omega = \frac{1}{2T} .$$

In terms of B_{eq},

$t_r = 0.45/B_{eq}$

$I_{max} = 2\ B_{eq}$

$$|H(f)|^2 = \frac{\sin^2 \left(\frac{\pi}{2} \frac{f}{B_{eq}} \right)}{\left(\frac{\pi}{2} \frac{f}{B_{eq}} \right)^2} .$$

APPLICATION NOTE: The constant time integrator can be made to produce output samples with a single integration by using an integrate, sample and dump technique. Ideally, then, a continuous constant time integration can be made by subtracting the input delayed by T from the present input before integration. Thus,

$$O(t) = \frac{1}{T}\int_{-\infty}^{t} f(x) - f(x-T)\,dx$$

$$= \frac{1}{T}\int_{-\infty}^{t} f(x) - \frac{1}{T}\int_{-\infty}^{A-T} f(x)\,dx$$

$$= \frac{1}{T}\int_{t-T}^{t} f(x)\,dx \quad .$$

The practicality of implementing the delay, however, severely limits this application.

B4.0 RECTANGULAR LOW PASS FILTER

This filter is physically unrealizable but represents the limit of achievement for absolute rejection of frequencies above a certain minimum. Thus, for a given noise equivalent bandwidth it gives the maximum rejection. The implementation can be approached with a Butterworth filter of a sufficiently high number of poles and appropriate phase compensating zeros. Zero phase shift was assumed for the ideal filter since this gives exact reproduction of signals within the band. The characteristics are:

$$H(\omega) = 1, \quad -2\pi B \leq \omega \leq 2\pi B$$

Where B is the single sided bandwidth.

$$H(\omega) = 0; \quad \omega < -2\pi B, \ \omega > 2\pi B$$

$$B_{eq} = \frac{1}{2\pi}\int_{o}^{2\pi B} 1\,d\omega = B$$

$$I(t) = \frac{1}{2\pi}\int_{-2\pi B_{eq}}^{2\pi B_{eq}} e^{j\omega t}d\omega$$

$$= \frac{1}{2\pi} \cdot \frac{2 \sin 2\pi B_{eq} t}{t}$$

$$I_{max} = 2B_{eq}$$

$$R(t) = \frac{1}{2} + \frac{1}{\pi}\int_{o}^{2\pi B_{eq} t} \frac{\sin x}{x}\,dx$$

For purposes of comparison, R (t) must be numerically evaluated.

APPLICATION NOTE: It should be noted that both I (t) and R (t) show responses for negative time. This is due to the physically non-realizability of the filter. A real filter could not have zero phase shift at all frequencies. Linear phase shift filters can be built, however, showing a constant time delay over the zero shift case. A reasonable approximation to a rectangular filter should be realized for a time delay equal to the last zero crossing of the ideal waveform in negative time. From the calculated value of R (t), this occurs at $t = -.309/B_{eq}$. The plot (in Section 4.5.2) of R (t) is shifted to the right by this amount, indicating physical realizability. The resulting plot shows a zero to 90 percent rise time of

$$t_r = 0.54/B_{eq}$$

INDEX

NOTES

NOTES